НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК 004.453

До захисту допущено:

Завідувач кафедри

_____ Ігор ПАРХОМЕЙ

«___»_____2020 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Інформаційне забезпечення робототехнічних систем»

зі спеціальності 126 «Інформаційні системи та технології»

на тему: «Система управління розумним будинком»

Виконав (-ла):
студент (-ка) ІІ курсу, групи ІК-91мп
Артем ЛЄМЕШКО _____

Керівник:
доцент, к.т.н.,
Олег ЛІСОВИЧЕНКО _____

Консультант з нормоконтролю:
доцент, к.т.н., доц.,
Віктор ПАСЬКО _____

Рецензент:
проф. каф. ОТ, д.т.н., проф.
Сергій СТІРЕНКО _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2020 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Ігор ПАРХОМЕЙ

«___»_____2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Лємешко Артему Дмитровичу

1. Тема дисертації «Система управління розумним будинком», науковий керівник дисертації Прізвище, ім’я, по батькові, науковий ступінь, вчене звання, затверджені наказом по університету від « 26 » жовтня 2020р. № 3132-с

2. Термін подання студентом дисертації _____23.11.2020 р._____

3. Об’єкт дослідження – розумний будинок.

4. Вихідні дані – відкриті дані інформаційних порталів, статті та підручники зарубіжних та вітчизняних авторів, існуючі прототипи програмних додатків.

5. Перелік завдань, які потрібно розробити – аналітичний огляд предметної області; огляд і порівняння існуючих рішень; дослідження ефективності використання протоколу z-wave; розробка системи та візуалізації управління розумним будинком.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу – два креслення, чотири плакати.

7. Орієнтовний перелік публікацій – 1 публікація.

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Перевірка на співпадіння	доцент Лісовиченко О.І.		
Нормоконтроль	доцент Пасько В.П.		

9. Дата видачі завдання 27.09.2019 р

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	01.09.2020 – 08.09.2020	
2	Аналіз існуючих прототипів та проблем	08.09.2020 – 15.09.2020	
3	Детальний опис предметної області	15.09.2020 – 30.09.2020	
4	Розробка та налаштування адаптерів	30.09.2020 – 07.10.2020	
5	Розробка інтерфейсу програмного забезпечення	07.10.2020 – 31.10.2020	
6	Маркетинговий аналіз стартап-проекту	31.10.2020 – 24.11.2020	
7	Попередній захист	25.11.2020	
8	Нормоконтроль	06.12.2020	
9	Перевірка на співпадіння	09.12.2020	
10	Захист	23.12.2020	

Студент Артем ЛЄМЕШКО

Науковий керівник Олег ЛІСОВИЧЕНКО

АНОТАЦІЯ

У роботі розглянуто проблеми управління автоматизованою системою розумного будинку, зручності налаштування автоматизованої системи. Порівнянно п'ять різних систем автоматизації, які підтримують Z-wave протокол.

Розроблено програмне забезпечення, а також інтерфейс за допомогою платформи iobroker. Встановивши адаптер VIS-interfase, створенно зручний інтерфейс управління домашніми пристроями, а також термостатами. Створенна можливість моніторингу ближніх країн із сітуацією зараження коронавірусом. Управляти та створювати нові речі в інтерфейсі программного забезпечення можна з будь-якого пристрою. Також встановлено один із головних адаптерів створення інтерфейсу, який має назву Material Design.

Ключові слова: інтерфейс управління, автоматизація, платформа iobroker.

Розмір пояснювальної записки – 100 аркушів, містить 49 ілюстрації, 25 таблиць, 6 додатків.

ABSTRACT

The problems of control of the automated system of the smart house, convenience of adjustment of the automated system are considered in the work. Relatively five different automation systems that support the Z-wave protocol.

Developed software as well as an interface using the iobroker platform. By installing the VIS-interface adapter, a convenient interface for controlling home devices and thermostats has been created. The possibility of monitoring neighboring countries with the situation of coronavirus infection has been created. You can manage and create new things in the software interface from any device. Also installed is one of the main adapters for creating an interface called Material Design.

Keywords: control interface, automation, iobroker platform.

Explanatory note size – 100 pages, contains 49 illustrations, 25 tables, 6 applications.

Пояснювальна записка

до магістерської дисертації

на тему: ***Система управління розумним будинком***

Київ – 2020 року

ЗМІСТ

СПИСОК СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

IoT – Інтернет речей

DG – Розподілене виробництво

SG – Інтелектуальна мережа

SH – Розумний будинок

ВСТУП

Розумні системи автоматизації будинку дозволяють користувачеві легше контролювати та відстежувати системи освітлення, безпеки, вентиляції та температури. Хоча ці системи забезпечували легке управління будинком залежно від команд, які користувач давав лише в ранній період, сьогодні, використовуючи алгоритми штучного інтелекту, розроблені для енергозбереження, домашньої безпеки, управління тощо, ці системи почали виконувати багато операцій автономно. Багато дослідників та дослідницьких компаній працюють у цій галузі та щодня розробляють нові алгоритми для створення нових продуктів. Алгоритми штучного інтелекту повинні постійно тестуватися та вдосконалюватися в різних умовах, щоб нормально функціонувати та виконувати бажану функцію. Виконання тестів алгоритму в реальних умовах спричиняє багато проблем і знижує ефективність з точки зору досліджень. У подібних ситуаціях в розумних будинках та різних дисциплінах дослідження з розробки та вдосконалення алгоритмів проводяться в модельованих середовищах. Однак, можливо, не вдасться включити всі змінні реального життя в середовища моделювання. Враховуючи всі ці ситуації, у цьому дослідженні розроблено гібридне моделювання розумного будинку, яке запускає реальні та віртуальні розумні будинки для тестування алгоритмів штучного інтелекту, розроблених для використання в розумних будинках. Справжня система розумного будинку вперше розроблена та встановлена в приміщенні для гібридного моделювання. Потім розробляється змодельований будинок із бажаною кількістю кімнат, бажаною кількістю компонентів розумного будинку з різними завданнями для цього будинку. Кімната, в якій працює справжня система розумного будинку, перетворюється на кімнату у віртуальному будинку, змінюючи коди моделювання. Так само реальна людина, яка користується справжнім розумним будинком, стає мешканцем, який живе у віртуальному домі. Крім того, віртуальні мешканці створені для використання справжньої системи розумного будинку. Таким чином, реальний та віртуальний розумний дім та реальні та віртуальні мешканці живуть разом у гібридному моделюванні. Нарешті, гібридне моделювання працює плавно за різних умов протягом двох місяців за допомогою різних алгоритмів штучного інтелекту.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Об'єкт та предмет дослідження

Об'єкт дослідження: Розумний будинок.

Предмет дослідження: Інтерфейс контролю життєдіяльності людини в середовищі розумного будинку.

Мета дослідження: Адаптація та моделювання роботи усіх пристроїв розумного будинку, перевірка на сумістність приладів розумного будинку, а також комфортне управління та моніторинг усієї системи розумного будинку.

Наукова новизна: Створення умов забезпечення життєдіяльності для кожного з жителів будинку (квартири), а також забезпечити систему охорони за допомогою.

Актуальність: Автоматизація управління розумним будинком, а також комфортний інтерфейс управління розумним будинком, на сьогоднішній день дуже важливе питання, бо інтерфейси створенні компаніями, не дуже зручні у користуванні людиною, яка не дуже розуміє як упправляти та створювати процесси автоматизації. Головною метою роботи є створення графічного інтерфейсу системи управління розумним будинком таким чином, щоб кожен користувач, міг створювати та бути впевненим в користування системою розумного будинку.

1.2 Огляд існуючих рішень

1.2.1 Гібридна система автоматизації

Процес розробки гібридного моделювання розумного будинку складається з трьох етапів. Спочатку була розроблена справжня система розумного будинку, а потім розроблено моделювання відповідно до цієї архітектури розумного будинку та структури даних. Нарешті, ці дві системи були об’єднані як гібридна концепція для одночасної роботи в реальному часі.

1.2.1.1 «Справжній» розумний дім

В рамках дослідження, по-перше, був розроблений хмарний розумний дім, який може контролювати температуру, світло та розетки кімнати через Інтернет та реєструвати всі виконані операції. Цей розумний будинок складається з компонента термостата, який контролює температуру, розетки, що контролює датчики електрики, температури, вологості та освітлення, кімнатного контролера, який контролює всі ці компоненти, та хмарного сервера, що дозволяє керувати та контролювати через веб-інтерфейс. Загальна робота системи може бути виражена наступним чином: користувач може бачити дані із системи розумного дому через веб-інтерфейс та керувати компонентами. Іншими словами, коли користувач хоче увімкнути лампу, вимкнути розетку або відрегулювати термостат, вони виконують цей процес через веб-інтерфейс, який є контрольною точкою для системи. Веб-інтерфейс представлений на рис. 1. Ця веб-сторінка працює на програмному забезпеченні веб-сервера (Apache), встановленому на центральному сервері.

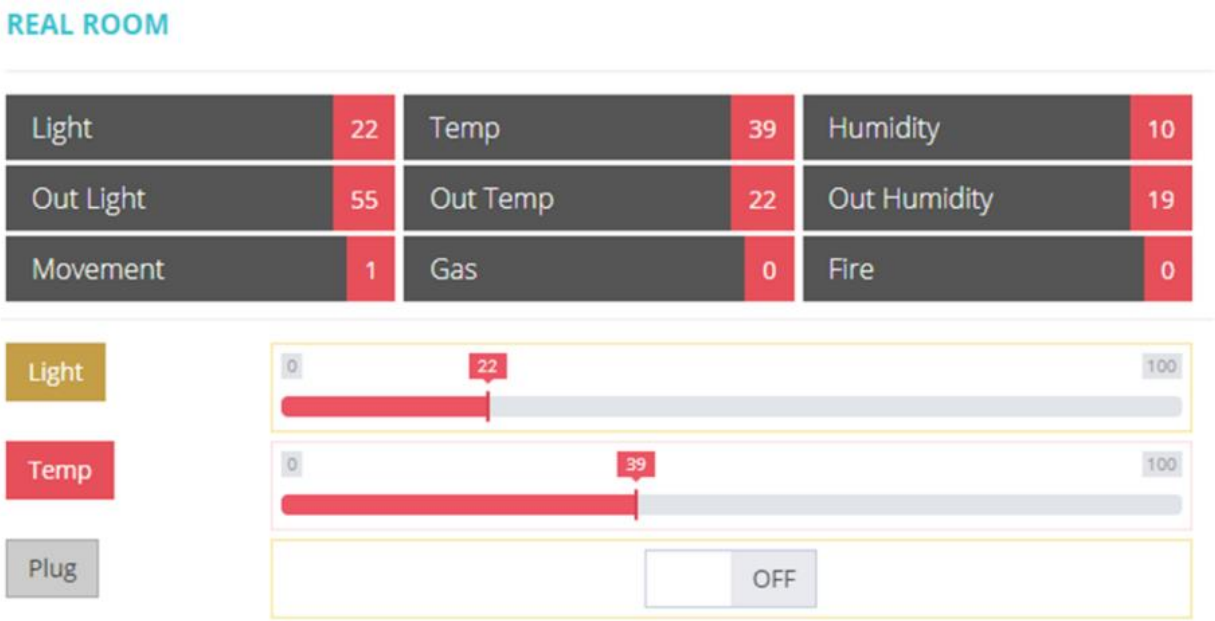


Рисунок 1.1 – Веб-додаток «справжнього» розумного будинку.

Центральний сервер - це серверний комп'ютер, який постійно працює, підключений до Інтернету і може знаходитись де завгодно фізично (у поточній системі сервер знаходиться в серверній кімнаті університету). Цей сервер передає транзакції, здійснені користувачем у веб-інтерфейсі, на відповідний контролер кімнати, підключений через Інтернет. Веб-інтерфейс, що працює на центральному

сервері, сумісний з мобільними пристроями, використовуючи мови HTML, CSS та Javascript. Тому мобільний додаток не розроблявся. Коли користувачі вперше входять у веб-інтерфейс, вони входять в систему. Після входу в систему можна побачити екран, на якому відображається інформація про температуру, вологість та освітлення приміщень, а також регулювати компоненти розумного будинку, такі як термостат та лампа (див. Рис. 1.1). Сервер MQTT працює на сервері, щоб надсилати транзакції до розумного будинку та відображати дані датчиків із розумного будинку на веб-інтерфейсі. MQTT - це широко використовуваний міжмережевий (M2M) протокол на основі повідомлень в Інтернеті. Сервер виконує операції, пов’язані з розумним будинком, через підключений до нього контролер кімнати. Крім того, система управління базами даних MySQL, що працює на сервері, реєструє транзакції, здійснені користувачем, дані, отримані від датчиків, інформацію про користувача та пристрій, а також будь-які інші дані, що використовуються в процесах штучного інтелекту.

У кожній кімнаті розумного будинку є контролер кімнати. Кімнатні контролери передають запити від веб-інтерфейсу системи через сервер до термостата, освітлення та компонентів розеток. Ці компоненти відповідають їхнім вимогам. Крім того, кімнатний контролер передає дані, які він збирає від підключених датчиків, на сервер.

Компонент термостата - це тип регулюючого пристрою для підтримання постійної температури до бажаної міри. Роль термостата в дослідженні полягає у підтримці температури кімнати, де він знаходиться, постійною на рівні, визначеному користувачем. Подібним чином, компонент освітлення регулює яскравість лампи в кімнаті відповідно до заданої користувачем яскравості. Компонент розетки дозволяє включати і вимикати розетку в кімнаті. Всі ці компоненти бездротово підключені до контролера кімнати через Bluetooth. Таким чином, його можна використовувати де завгодно в кімнаті. Автором гібридної системи моделювання управління розумним будиком є Сабрі Бічакчі та Хусейн Гюнес.

1.2.1.2 Компоненти інструментів для створення розумного будинку

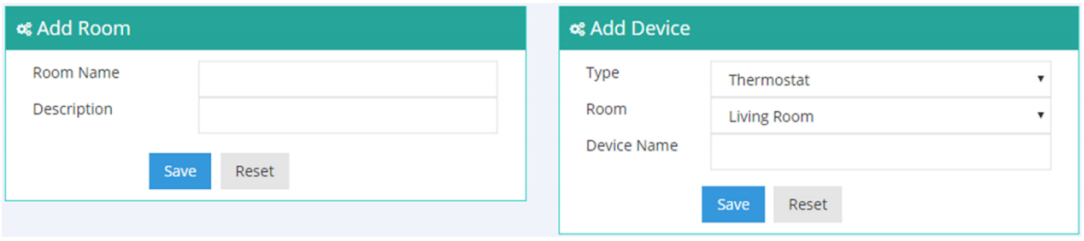


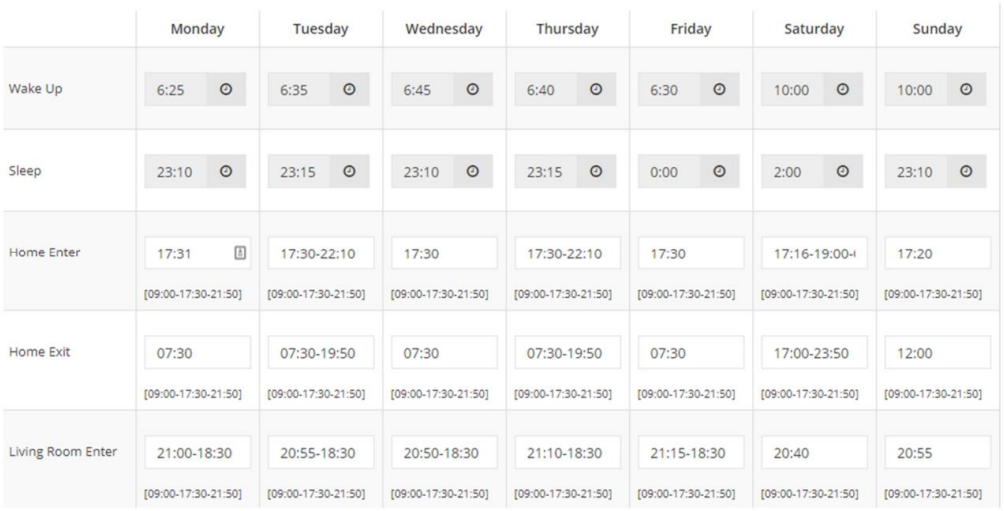
Рисунок 1.2 – Вікна додатку, котрі дають змогу добавити кімнату та пристрій у систему розумного будинку.

У дослідженні був розроблений інструмент, який може створити бажану кількість кімнат, що дозволяє віртуально створювати будинок. За допомогою цього інструменту кімната створюється шляхом введення лише назви кімнати та короткого опису. Пізніше був розроблений інструмент для додавання компонентів розумного будинку до створених кімнат. За допомогою цього інструменту компоненти розумного будинку можна додати до кімнати, ввівши тип пристрою, кімнату та назву пристрою. Знімок екрана інструменту для додавання приміщення та пристрою представлений на рис. 1.2.

1.2.1.3 Мешканці віртуального будинку та сценарії

Інструмент створення віртуального індивіда був розроблений для того, щоб створити віртуальних людей, які замінять реальних людей у моделюванні. Під час створення особи за допомогою цього інструменту вводяться коротке ім’я віртуальної особи, яке видно під час моделювання та коротка інформація, яка ідентифікує особу.

Після визначення віртуальної людини був розроблений інструмент, за допомогою якого можна визначити життєві сценарії для людини. За допомогою цього інструменту, коли віртуальна людина буде входити і виходити з кімнати, коли вона виходитиме і повертатиметься додому, який розумний-пристрій буде вмикати або вимикати людина, коли пристрій використовується, яким буде діапазон значень та бажаний температура та яскравість у середовищі проживання визначаються щотижня. На рис. 1.3 показаний розділ екрану створення сценарію.



	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Wake Up	6:25	6:35	6:45	6:40	6:30	10:00	10:00
Sleep	23:10	23:15	23:10	23:15	0:00	2:00	23:10
Home Enter	17:31 [09:00-17:30-21:50]	17:30-22:10 [09:00-17:30-21:50]	17:30 [09:00-17:30-21:50]	17:30-22:10 [09:00-17:30-21:50]	17:30 [09:00-17:30-21:50]	17:16-19:00-I [09:00-17:30-21:50]	17:20 [09:00-17:30-21:50]
Home Exit	07:30 [09:00-17:30-21:50]	07:30-19:50 [09:00-17:30-21:50]	07:30 [09:00-17:30-21:50]	07:30-19:50 [09:00-17:30-21:50]	07:30 [09:00-17:30-21:50]	17:00-23:50 [09:00-17:30-21:50]	12:00 [09:00-17:30-21:50]
Living Room Enter	21:00-18:30 [09:00-17:30-21:50]	20:55-18:30 [09:00-17:30-21:50]	20:50-18:30 [09:00-17:30-21:50]	21:10-18:30 [09:00-17:30-21:50]	21:15-18:30 [09:00-17:30-21:50]	20:40 [09:00-17:30-21:50]	20:55 [09:00-17:30-21:50]

Рисунок 1.3 – Інструмент щотижневого введення сценарію для віртуальної особи.

Хоча люди мають загальні звичні звички, ніхто не робить точно одне і те ж щодня в один і той же час. Наприклад, людина повертається з роботи щодня близько 17.00, не точно о 17.00 щодня. Часові відхилення трапляються в цих загальних рутинних завданнях. З огляду на ці ситуації, значення діапазону відхилень можуть бути визначені для часу входу-виходу та інших налаштувань, зроблених для кожної віртуальної людини відповідно до реального життя, за допомогою інструменту введення сценарію особи.

1.2.1.4 Інфраструктура моделювання

Після створення розумного будинку, компонентів, мешканців та сценаріїв було розроблено програмне забезпечення, яке дозволяє працювати з моделюванням відповідно до цих даних, з мовою програмування Javascript для роботи в середовищі виконання Node.js. Використовуючи PM2 (розширене програмне забезпечення для керування процесами Node.js), також розроблене для виконання Node.js, моделювання може бути перезапущено автоматично у разі можливих збоїв, перезапуск системи і, таким чином, забезпечується безперервна робота [24].

Веб-інтерфейс був створений за допомогою мов програмування HTML, CSS та JavaScript для моніторингу моделювання та втручання при необхідності. За допомогою цього веб-інтерфейсу розумний дім, компоненти та індивідууми відображаються у висоті пташиного польоту та в реальному часі (див. Рис. 1.4).

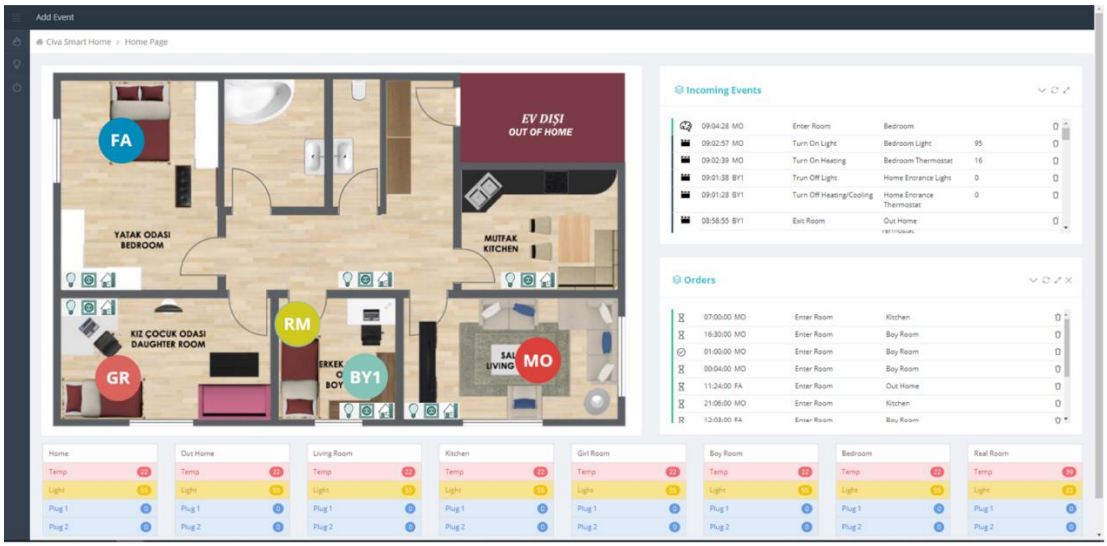


Рисунок 1.4 – Веб-інтерфейс моделювання.

На рис. 1.4 кола з висоти пташиного польоту розумного будинку з двома буквами представляють віртуальних людей. Ці кола автоматично відображаються у відповідній кімнаті, коли люди переходять із кімнати в кімнату. Їх положення в кімнаті виробляється випадковим чином, щоб не перекриватися. Значки ламп, температури та розетки в кімнатах символізують типи компонентів розумного будинку в цій кімнаті.

Праворуч від інтерфейсу розташовані дві таблиці. У наведеній вище таблиці наведено майбутні транзакції, тобто горизонт подій, які люди виконуватимуть за визначеними сценаріями. У наведеній нижче таблиці наведені розпорядження, що даються менеджером поза сценарієм. У маленьких таблицях внизу сторінки відображаються дані про температуру та світло з усіх кімнат розумного будинку та стан розеток у кімнатах.

У верхній та лівій частині екрана є меню, крім візуальної частини в центрі. Меню з лівого боку містять блоки управління, де компонентами розумного будинку в кімнатах можна керувати безпосередньо. Звідси поточне налаштування температури, освітлення або компонента розетки у бажаній кімнаті можна миттєво побачити та змінити. На рис. 6 показано меню, де можна налаштувати температуру для всіх кімнат.

Дії можна додати поза сценаріями за допомогою меню «Додати подію» вгорі. Наприклад, із меню «Додати подію», зображеного на рис. 7, можна додати особу, яка переїжджає з однієї кімнати в іншу або змінює налаштування компонента розумного

будинку, вказавши час, і вона може автоматично виконуватися системою, коли приходить час.

1.2.2 Інструмент автоматизації zVirtualScenes

zVirtualScenes - це програмний контролер сцен ZWave, де ви можете створювати власні сцени ZWave для відтворення через графічний інтерфейс, команди HTTP або смартфон.

Рисунок 1.5 – Інтерфейс додатку zVirtualScenes на усіх платформах.

– Отримайте повний контроль над своїм сумісним пристроєм ZWave за допомогою простого графічного інтерфейсу.
– ZVirtualScenes відстежує ваші пристрої ZWave щодо змін стану і може надсилати сповіщення на Jabber / Google Talk. Наприклад, отримуйте повідомлення за допомогою миттєвого повідомлення, якщо температура опускається нижче настроюваної межі. Отримуйте сповіщення про зміну рівня перемикача або ввімкнення термостата.
– Керуйте своїми пристроями ZWave зі свого смартфона (Android та IPhone).
– Використовуйте zVirtualScenes Mobile повноцінний мобільний додаток Sencha на основі HTML5 / JS, щоб повністю контролювати свої пристрої з будь-якого місця.
– Легко створюйте власні сцени, керуючи термостатами, перемикачами та іншим одним простим дією. Активуйте їх за допомогою смартфона, HTTP або графічного інтерфейсу.

– Повністю протестований за допомогою дистанційного терморегулятора Trane ZWave.
– Використовуючи плагін Web API, ви можете взаємодіяти майже з усіма аспектами zVirtualScenes, використовуючи RESTful API. Створюйте сцени, перейменовуйте пристрої, виконуйте команди тощо.
– zVirtualScenes має плагін-систему з повним API. Розробники можуть легко створювати власні плагіни у Visual C #.

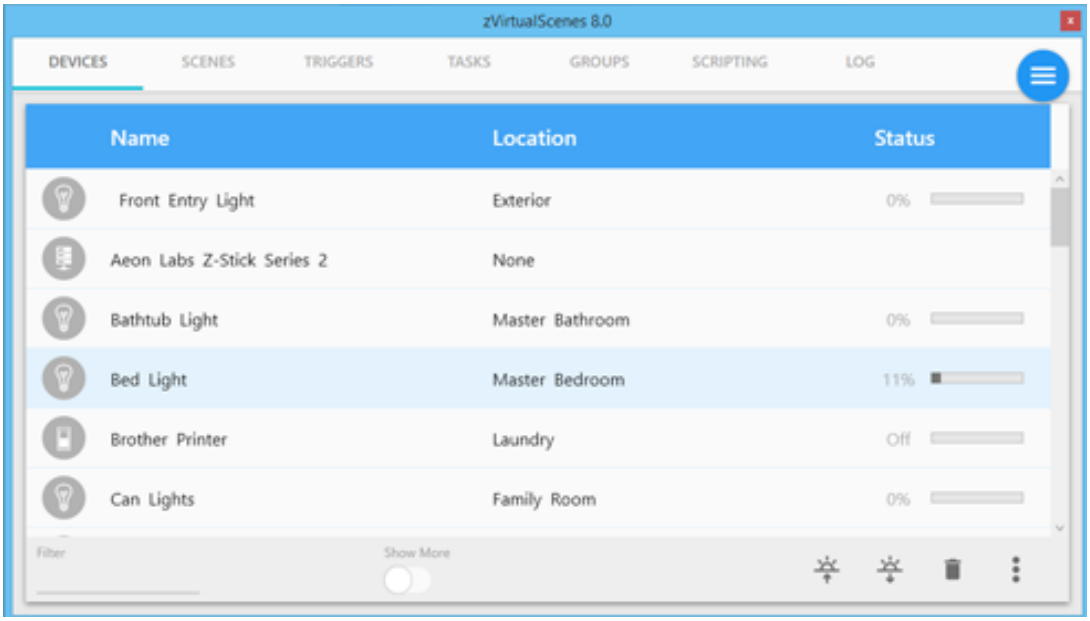


Рисунок 1.6 – Інтерфейс додатку zVirtualScenes управління пристроями.

Станом на версію 3.5 ми дозволили підтримку JavaScript, який буде використовуватися в Scenes як команди. Ця сторінка надасть вам огляд того, як використовувати цю нову та захоплюючу функцію.

Основний доступ до редагування сценаріїв можна знайти в головному меню, командах, пункті меню Додати / редагувати JavaScript.

Для того, щоб пов'язати конкретний сценарій зі сценою, відредагуйте свою сцену, і внизу праворуч ви можете натиснути "Додати команди", "Додати команду JavaScript", що дозволить вам вибрати сценарій, який ви раніше редагували.

Існує два методи використання сценаріїв. Перший - це безпосереднє редагування сценарію в інтерфейсі та виконання всіх необхідних дій. Базову бібліотеку, яку ми використовуємо, можна знайти тут: http://jint.codeplex.com. Зверніть увагу, що у вас є повний доступ до просторів імен .NET, тому це буде дійсним:

logInfo (System.DateTime.Now);

Другим методом використання скриптів буде функція require (). В інсталяційній папці програми ви знайдете папку "сценарії". Помістіть тут файли .js та імпортуйте їх у свій основний сценарій, наприклад:

require("gmail.js");

1.2.3 Система автоматизації Ago Control

Ago Control - це система управління пристроєм. Мета - надати повне рішення для автоматизації будинку. Його також можна використовувати в інших сферах, таких як сільське господарство. тому управління використовує шину повідомлень AMQP Enterprise Message як вихідну інформацію, легкий протокол, читається людьми та машинами, сучасну та модульну архітектуру, хмарні функції тощо!

Ago Control має чудову продуктивність, а також працює на вбудованих пристроях, таких як Raspberry Pi, та кількох обчислювальних штекерах, таких як Sheevaplug, Guruplug та Pogoplug.

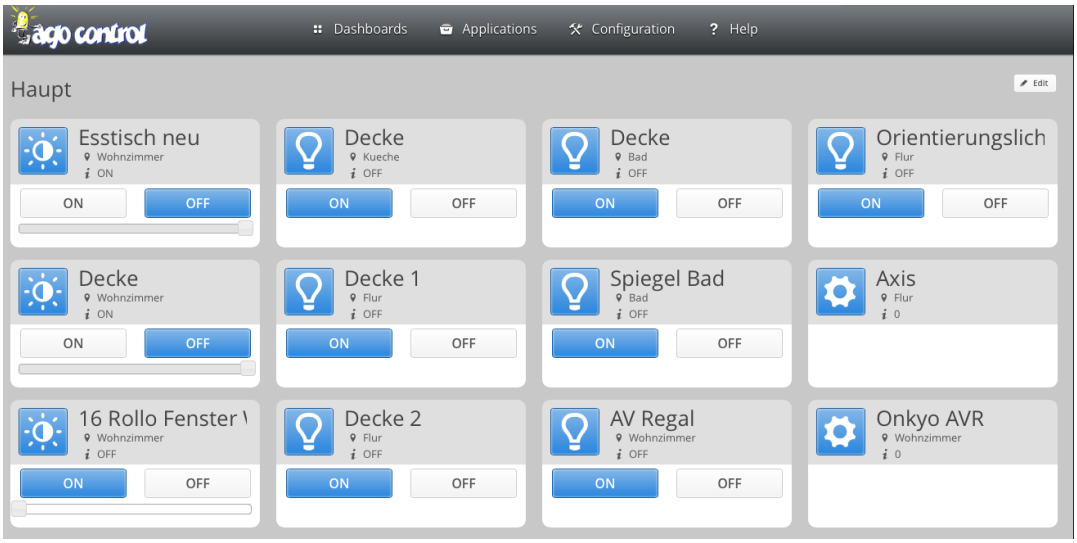


Рисунок 1.7 – Інтерфейс додатку Ago Control.

Він має підтримку багатьох пристроїв і протоколів, таких як Z-Wave, KNX, трансивери 433 МГц, EnOcean, X10, 1wire, ATC Asterisk, Dreambox / Enigma2, Onkyo eISCP AVR, світлодіодний диммер Chromoflex USP3 RGB, блок розподілу живлення APC (PDU), Інтерфейси DMX через OpenLightingArchitecture (OLA), телевізори Phillips (Jointspace), Arduino Firmata, споживчий інфрачервоний бластер IRTrans Ethernet, міст Ethernet Kwikwai HDMI CEC, контролер зрошення Rain8net, підтримка

веб-камери та підтримка BlinkM LED. тому контроль легко розширити, і в ньому зростає перелік драйверів пристроїв, що подаються користувачем.

Додаткову допомогу можна знайти на попередніх форумах управління та / або на IRC через freenode.net у каналі #agocontrol.

Основні компоненти. Перелік різних компонентів, які контролюють і діють на системні змінні та різні пристрої введення, налаштовані в попередній системі управління:

- Resolver - центральний компонент, який обробляє "реєстрацію" пристрою та вирішує імена
- Таймер - компонент таймера попереднього контролю, який використовується для ініціювання подій, заснованих на часі
- Конфігурація подій - Компонент, який використовує визначені користувачем тригери, що відповідають певним критеріям, а потім діє на них, як визначено дією
- Конфігурація сценарію - Компонент, що використовується для групування команд керування тому для створення простих або складних дій, які можуть бути викликані подіями, з інформаційної панелі або з планів поверху
- RPC Interface - Серверний компонент веб-адміністратора ago control
- Веб-адміністратор - Налаштування та моніторинг пристроїв, подій та сценаріїв, використовуючи веб-адміністратор контролю за минулим часом

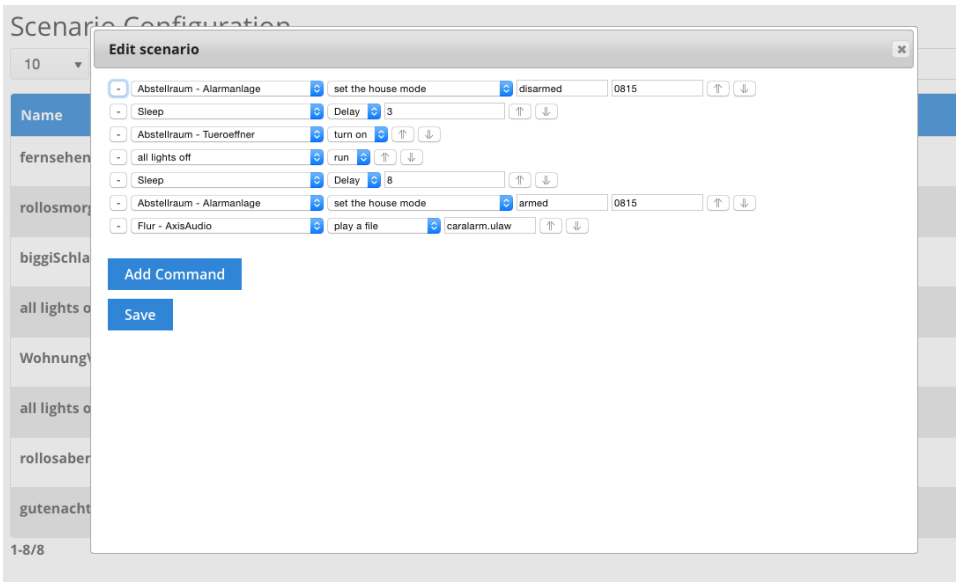


Рисунок 1.8 – Інтерфейс створення сценаріїв Ago Control.

- Реєстратор даних - реєстратор даних реєструє всі події, які надсилаються пристроями
- Blockly - простий спосіб створення дій без будь-яких навичок роботи з комп’ютером (або трохи) за допомогою побудови блоків
- SecuritySystem - функції сигналізації

1.2.4 Система автоматизації ioBroker

ioBroker - це програмне рішення для інтеграції різних систем розумного будинку, які залишилися б автономними рішеннями без ioBroker, в одній загальній системі. Таким чином, ioBroker є інтеграційною платформою для Інтернету речей. Система ioBroker має модульну структуру. Велика кількість адаптерів забезпечує зв'язок з більш ніж 200 різними платформами від A для Alexa до Z для запису часу.

Будь то інтеграція комерційних продуктів практично з усіх сфер життя або інтеграція внутрішньо генерованого рішення - ioBroker робить все можливе.

Той, хто займається домашньою автоматизацією, рано чи пізно виявить, що системи часто недосконалі. У кожної системи є свої сильні і слабкі сторони. Таким чином, ioBroker є кросплатформним. У будь-який момент можлива паралельна робота з існуючими рішеннями. Можна використовувати ефекти синергії і об'єднати найкраще з світів.

Сам ioBroker доречний практично на всіх платформах. ioBroker можна встановити в Windows, Linux, OSX або як образ Docker. Попередньо налаштовані установчі образи позбавляють користувача від роботи по установці.

Віддалене управління локально встановленою системою ioBroker можливо 24/7 для користувача або для системних інтеграторів через додатковий хмарний доступ. Контроль доступу може бути вільно налаштований користувачем з використанням користувачів і груп.

Якщо з часом повинні бути підключені інші системи розумного будинку, вони можуть бути реалізовані користувачем в будь-який час за допомогою додаткових адаптерів під час роботи. Сам ioBroker також масштабується. Кілька серверів ioBroker можуть бути підключені для формування системи Mutihost. Можна навіть

комбінувати платформи операційних систем і з'єднувати одноплатні комп'ютери SoC з великими багатоядерними серверами. Для систем з високими вимогами до продуктивності в якості опції може бути інтегрована Redis, особливо швидка база даних.

Додаткове програмування виконується за допомогою JavaScript, мови сценаріїв, який безперервно розвивається з 1995 року. Цьому легко навчитися, тому нові вимоги можна буде швидко реалізувати. Це дозволяє кожному внести свій вклад в ioBroker, а також можуть бути реалізовані індивідуальні вимоги.

Для новачків в програмуванні доступний варіант «Blockly», який дозволяє швидко отримати результати самостійно без великих знань програмування за допомогою «перетягування».

З VIS ioBroker надає потужний інструмент для створення індивідуальної візуалізації. Поточні значення від датчиків можуть бути представлені графічно так само, як історичні процеси. Прямі зображення з камер спостереження, установка системи сигналізації, систем опалення та кондиціонування - майже все, що тільки можна уявити, теж може бути реалізовано.

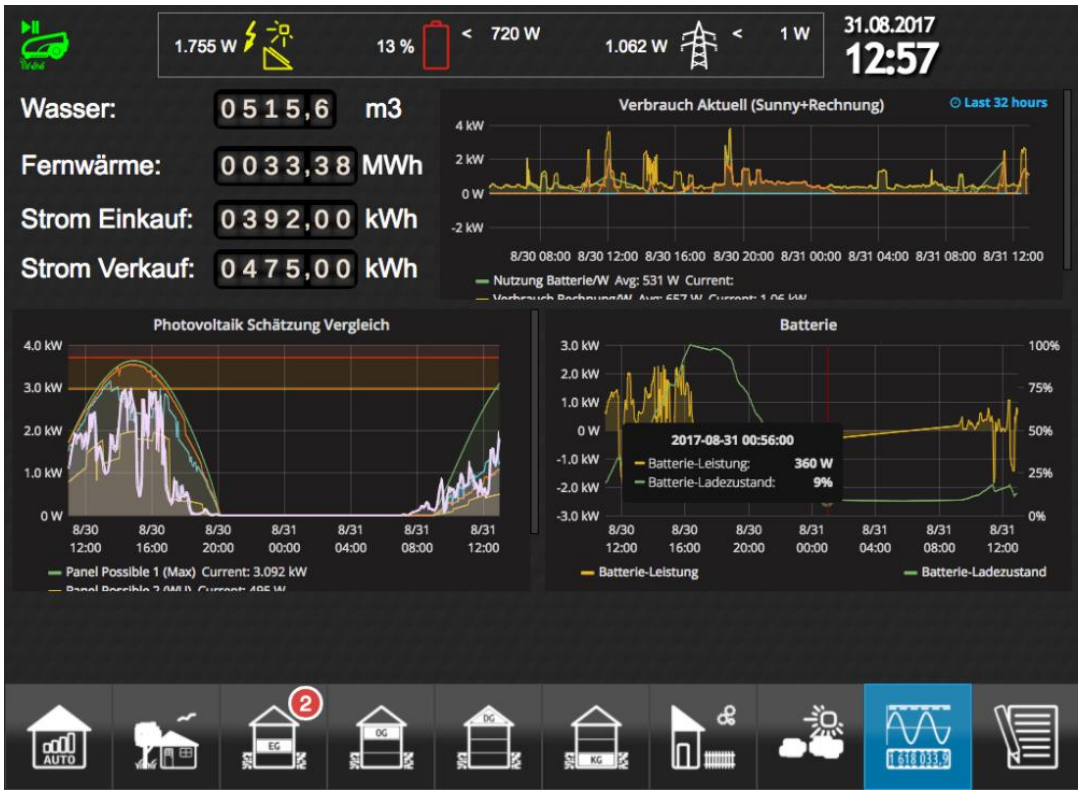


Рисунок 1.9 – Один із прикладів інтерфейсу додатку ioBroker.

Користувач має максимальну свободу дизайну. Готові модулі для зручного використання допомагають користувачеві в цьому. Але можливо не тільки відображення інформації. Пристроями також можна швидко керувати за допомогою інтерфейсу візуалізації. Поверхня може бути адаптована до найвибагливішим кінцевим пристроям - від смартфонів до настінних планшетів з сенсорними функціями і до персональних комп'ютерів - все можна реалізувати простим перетягуванням.

Прості готові призначені для користувача інтерфейси можна швидко реалізувати за допомогою адаптера матеріалу або HabPanel.

Рисунок 1.10 – Один із видів інтерфейсу створенних за допомогою ioBroker.

ioBroker - це чисте програмне рішення для підключення різних систем IoT до повної системі. Відповідно, центральний офіс (шлюз / інтерфейс) також потрібно для кожної системи, щоб мати можливість інтегрувати свої пристрої.

В особливих випадках така панель управління може бути змодельована програмним забезпеченням або як апаратне забезпечення (USB-накопичувач або подібне), заражене сервером ioBroker.

ioBroker має модульну структуру. Ці модулі називаються ioBroker Adapter. Є більше 250 адаптерів для підключення різного устаткування або інтеграції різноманітної інформації, такої як погода, календар і т. Д.

Тому в установці повинні бути встановлені тільки ті адаптери, які необхідні для ваших індивідуальних потреб. Це економить місце для зберігання та обчислювальну потужність.

Для кожного адаптера створюються так звані екземпляри. Це «робочі версії» адаптерів. Залежно від адаптера може бути створено будь-яку кількість екземплярів, щоб відрізняти різні підсистеми або різні області завдань один від одного.

Відповідна конфігурація має місце в цих випадках.

Особливістю ioBroker є те, що завдання також можуть бути розподілені на кілька серверів ** **. У такому випадку говорять про многоузловой системі. Причинами поділу можуть бути просторовий розподіл або розподіл потужності.

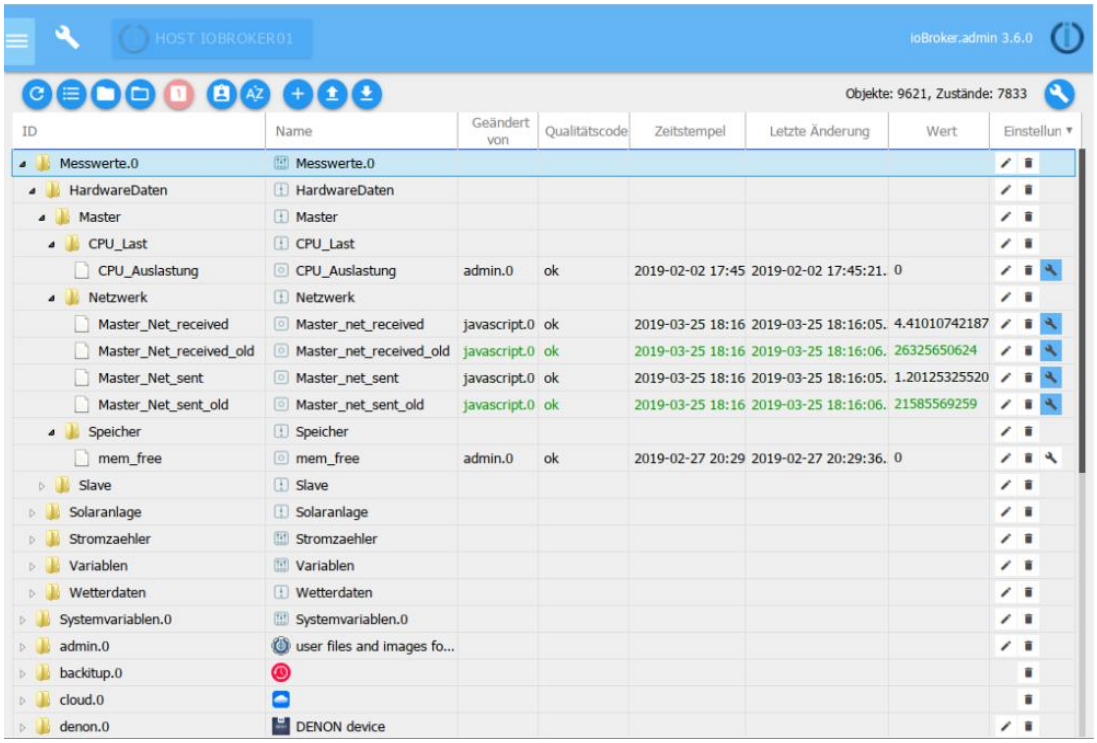


Рисунок 1.11 – Панель адміністратору ioBroker.

Вимоги до обладнання

Сервер ioBroker можна встановити практично на будь-якому обладнанні. Єдиною умовою є наявність поточної версії NodeJS для відповідної операційної системи.

Для більшої установки рекомендується ОЗУ об'ємом не менше 2 ГБ. Для тестування досить Raspberry Pi 2/3 з 1 ГБ оперативної пам'яті, так як в якості веденого пристрою для окремих адаптерів в середовищі з декількома хостами іноді досить навіть невеликих миникомпьютеров.

Програмне забезпечення

ioBroker управляє даними в базі даних. Відповідно структура даних організована.

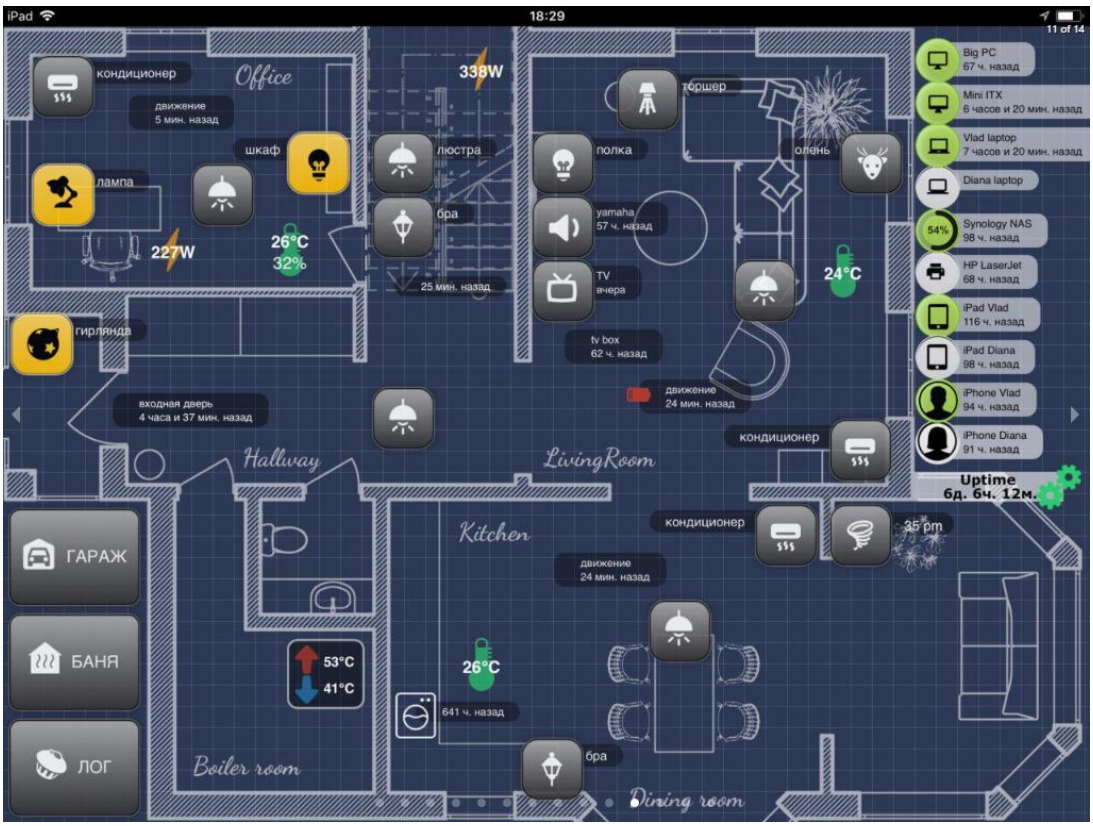


Рисунок 1.12 – Один із прикладів інтерфейсу додатку ioBroker.

Кожен адаптер має так зване простір імен, яке містить всі дані про екземпляр адаптера. Відповідно, ім'я простору імен, наприклад: AdapterName.0

У цьому діапазоні ioBroker створює пристрої, їх канали і точки даних зі своїми значеннями (станами).

Цей приклад є самостійно створений простір імен для ваших власних метрик.

1.2.5 Система автоматизації Domoticz

Domoticz - це дуже легка система домашньої автоматизації, яка дозволяє контролювати та налаштовувати різні пристрої, включаючи ліхтарі, вимикачі, різні датчики / лічильники, такі як температура, кількість опадів, вітер, ультрафіолетове (УФ) випромінювання, використання / виробництво електроенергії, споживання газу, споживання води і багато іншого. Сповіщення / попередження можна надсилати на будь-який мобільний пристрій.

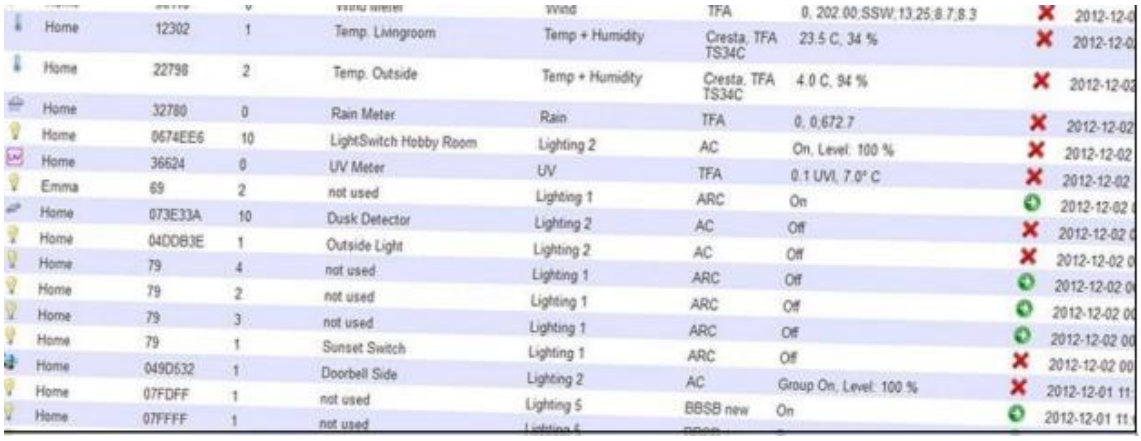


Рисунок 1.13 – Інтерфейс додату редактору пристроїв Domoticz.

Z-Wave протокол у системі автоматизації Domoticz

Domoticz використовує бібліотеку з відкритим кодом під назвою "OpenZWave". Це зазвичай пишеться "OZW". Оскільки протокол Z-Wave не є "безкоштовним", не всі пристрої або функції відомі в OZW. Більшість пристроїв повністю працюють (Fibaro, BeNext, Aeon для найвідоміших), інші просто посилаються без дії (fortrezz та інші, ...).

Щоб бути придатним для використання в Domoticz, спочатку потрібно перевірити, чи придатні ваші пристрої Z-Wave з OZW. Якщо ні, дотримуйтесь інструкцій щодо додавання пристроїв, щоб включити ваш у базу даних пристроїв OpenZWave.

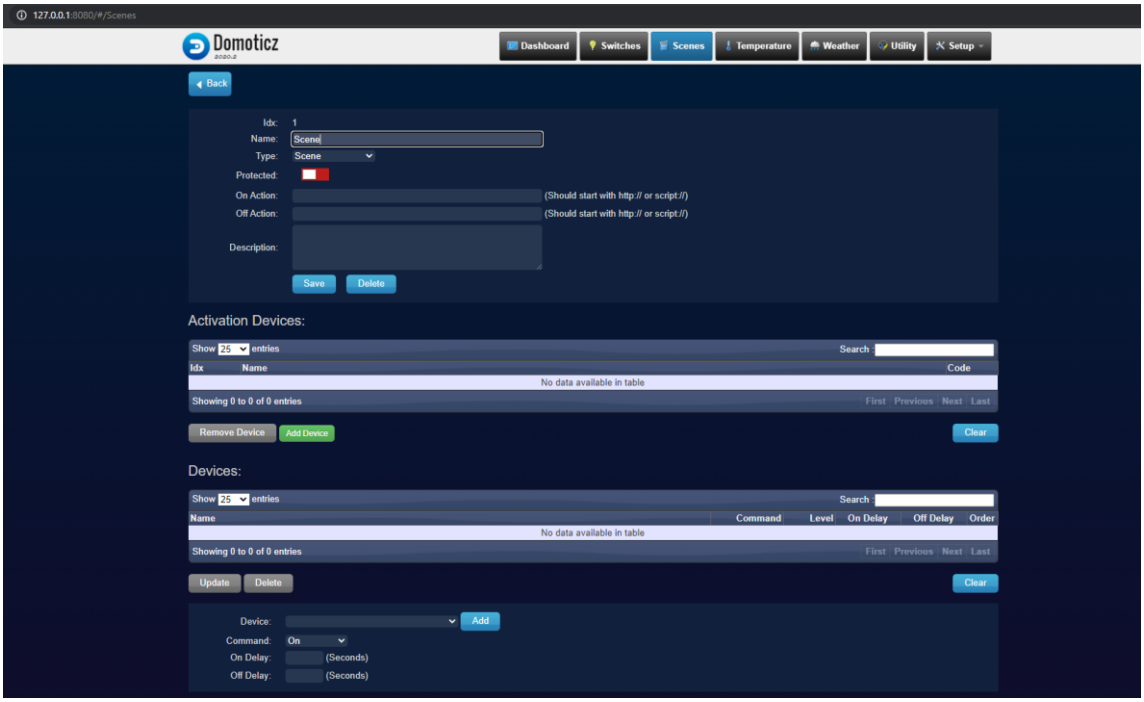


Рисунок 1.14 – Інтерфейс додату редактору сценаріїв Domoticz.

Інформація про мережу та групи

Екран інформації про мережу Z-Wave доступний у розділі Налаштування> Апаратне забезпечення. Якщо в системі є контролер Z-Wave (наприклад, USB-накопичувач), біля нього є кнопка "Налаштування". Клацніть на це, і з’явиться список керованих вузлів. У верхньому правому куті на кнопці управління вузлом розміщена кнопка "групи та мережа". З'явиться екран вище. Зліва - хордова діаграма, яка показує, які вузли бачать один одного. Це особливість сітчастої мережі Z-Wave; деякі пристрої не бачать одне одного безпосередньо, але взаємодіють через проміжні вузли. Цей графік допомагає зрозуміти спосіб взаємодії мережі. Якщо мережевий контролер Z-Wave щойно запустився (через запуск або перезавантаження Domoticz), для відображення з'єднань знадобиться деякий час. Зачекайте трохи і поверніться на цей екран. Наведіть курсор миші на вузол, щоб відобразити лише з'єднання для цього вузла; це полегшує вивчення конкретних зв’язків.

Праворуч - таблиця груп. Він показує зареєстровані групи для кожного вузла, а також вузли, що входять до групи. Групи Z-Wave є важливою частиною функціонуючої системи; він працює навіть без активного Domoticz або контролера, тому добре розуміти, як вони працюють і як налаштовуються. Групи не є глобальними, але існують для кожного вузла. Кожен вузол має власну групу (групи), в якій ви можете розмістити інші вузли як прослуховувачі. Якщо два вузли (пристрої) мають групу 2, це не одна група. Натискання номера вузла показує можливість видалення вузла з цієї групи. Якщо в групі є символ плюса, у вузлі ця група ввімкнена. Клацніть символ плюс, щоб додати ще один вузол до цієї групи.

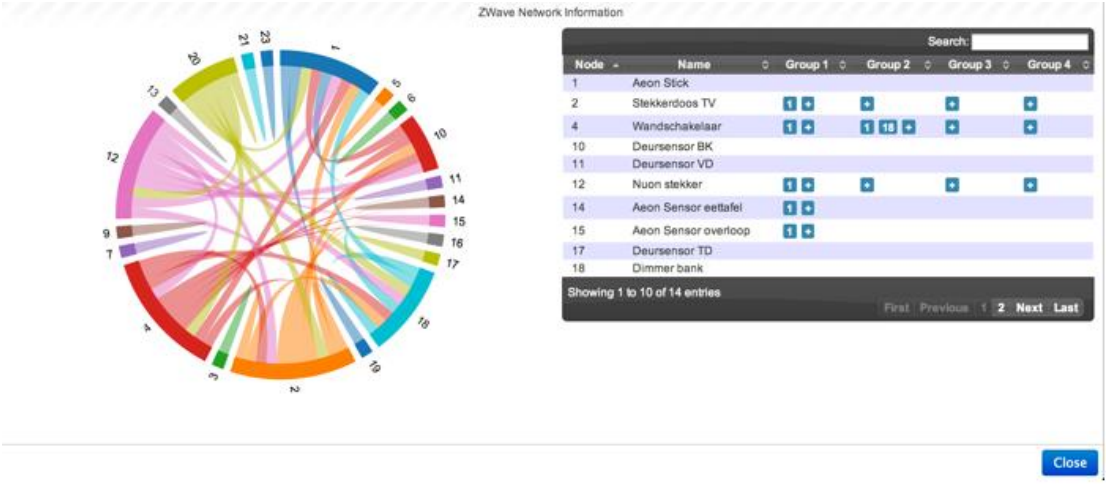


Рисунок 1.15 – Інформація про мережу та групи Domoticz.

Приклад: настінний вимикач tz66-d має дві лопаті (ліву та праву). Ліве весло фізично керує прикріпленою до нього лампою. Праве весло лише сигналізує про групи. Tz66-d має чотири групи:

Група 1: отримує сигнал при натисканні на ліве весло

Група 2: отримує сигнал при натисканні на праве весло

Група 3: отримує сигнал, коли двічі натиснуто право весло

Група 4: отримує сигнал, коли tz66-d наказано вмикати або вимикати.

Використовуючи таблицю груп, ви можете включити інший вузол (наприклад, лампу) в одну з груп tz66d. Поставивши його в групу 2, ви можете керувати ним за допомогою правого весла. Поставивши іншу лампу в групу 3, ви керуєте такою лампою подвійним натисканням на правому веслі тощо.

Групи також відіграють важливу роль у тому, коли вимірювальні прилади (наприклад, температура) надсилають свої дані. Деякі пристрої (наприклад, Aeon 4in1) можна налаштувати для відправки в групу 1 (де знаходиться контролер і, отже, Domoticz), а інші просто надсилають у фіксовану групу, для якої потрібно, щоб приймаючий пристрій було поставлено в цю групу.

Інформаційний екран у майбутньому буде розширено з більшою функціональністю. На даний момент у нього є найнеобхідніше;) Управління сценами поки що не існує, оскільки це можна зробити за допомогою самого Домотіча.

1.4 Постановка задачі

Основними задачами магістерської дисертації є:

- Дослідження актуальності та особливостей впровадження автоматизації в управління виробничими системами;
- Аналіз ефективності автоматизації в гнучких виробничих системах;
- Дослідження підходів до проектування автоматизованих систем;
- Аналіз структури автоматизованих систем;
- Дослідження можливостей інтеграції штучного інтелекту;
- Аналіз гнучких виробничих систем;
- Розробка програмного додатку згідно нижчеперерахованим вимогам.

Графічний інтерфейс є середовищем роботи користувача з даними, тому його розробка є одним з найважливіших моментів написання якісної конкурентоспроможної системи.

Отже, програмне середовище повинно відповідати таким вимогам:

- користувач повинен бути ініціатором всіх дій;
- програма повинна швидко реагувати на команди користувача;
- додаток повинен бути інтерактивним;
- надавати образне уявлення операцій, дій;
- доступність функцій самостійного введеня назв операцій;
- дружній та зрозумілий інтерфейс введення налаштувань;
- наявність функцій що упереджають випадкові дії зі сторони користувача;
- наявність функцій що дозволяють моделювати виробництво у 3D-середовищі;
- вибір об'єкта дій забезпечує доступ до засобів управління об'єктом, зміни положення об'єкта;
- наглядність подання важливої інформації при маніпулюванні об'єктами;
- однаковість методів роботи з системою;
- однаковість стилів та кольорів інтерфейсу користувача;
- наявність функцій збереження та завантаження налаштувань;
- доступність зворотньої інформації про хід процесу або режиму роботи;
- легкість освоєння і застосування;
- баланс між простотою і доступністю функцій і даних;

Висновки до розділу

У розділі визначено предмет та об'єкт дослідження, детально розглянуті існуючі аналоги створення систем автоматизації та візуалізації інтерфейсу користувача. Розглянуто існуючий спосіб рішення, а саме створення інтерфейсу за допомогою серверу на базі Apache, а також створення модулів управління на базі Arduino. Переглянуто чотири платформи систем автоматизації, котрі підтримують протокол розумного будинку Z-Wave: zVirtualScenes, Ago Control, ioBroker, Domoticz. Переглянувши переваги та недолікі кожної з платформ систем автоматизації, вибрано

ioBroker. Переваги ioBroker: існує багато рішень проблем синхронізації та інтеграції сторонніх протоколів розумних будинків, а також підтримка розробників адаптерів та візуалізації інтерфейсу користувачів.

РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Аналіз безпеки використання протоколу розумного будинку

Завдяки стрімкому технологічному прогресу міста стають розумнішими. Складові розумного міста включають розумний спосіб життя, розумне середовище, розумну мобільність, розумну економіку, розумне управління та розумних людей. Інфраструктура пов'язана з цими розумними технологіями, відома як Інтернет речей (IoT), Інтернет даних (IoD), Інтернет послуг (IoS) та Інтернет людей (IoP). Зокрема, оскільки IoT стає звичною технологією, важливими сферою турботи також стає цифрова безпека. Нещодавно технологія IoP постала у центрі досліджень, оскільки в розумних містах особиста інформація збирається за допомогою численних машин, потрібно забезпечити персоналізацію обслуговування приватних осіб. Однак існують серйозні проблеми щодо загрозі особистій інформації. Тому дослідження зфокусовані на особистій інформації, що стоїть поряд із безпекою людей, які живуть у розумних містах. У розумних містах приватні розумні будинки є основними джерелами збору даних. Оскільки розумні будинки стають основним компонентом розумних міст, існує потреба у дослідженні технологій їх застосування та кібербезпеки; тим часом, тонкощі збору даних із розумних будинків також досліджуються.

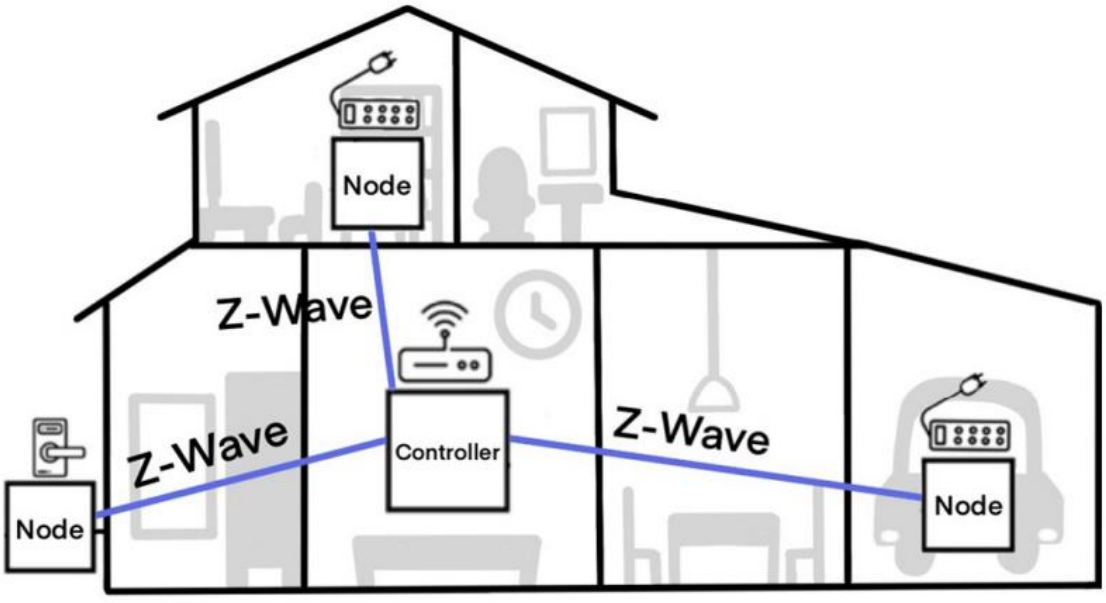


Рисунок 2.1 - Розумний будинок із Z-Wave

Різні пристрої все частіше вбудовуються в розумні будинки, щоб забезпечити жителям більш зручне середовище. За допомогою мобільних пристроїв користувачі можуть віддалено контролювати будинки, керувати газовими клапанами або відкривати замки дверей. Багато бездротових технологій, таких як Wi-Fi, ZigBee та Z-Wave, використовуються, щоб дозволити мешканцям розумних будинків дистанційно керувати розумними пристроями у своїх будинках. Ці мобільні пристрої та контролери використовують технології довгострокового розвитку (LTE) або Wi-Fi; однак, ZigBee та Z-Wave, які є технологією зв'язку короткого діапазону, використовуються в сенсорних мережах між пристроями в розумному будинку [1]. Z-Wave був розроблений датською компанією Zensys в 2001 році, і з тих пір він сертифікований для різних пристроїв альянсом Z-Wave. Незважаючи на те, що дослідження безпеки активно проводяться для інших існуючих технологій бездротового зв'язку, досліджень Z-Wave бракує, оскільки це приватний протокол. Тому буде детально проаналізовано протокол Z-Wave, який використовується в розумних будинках. Як показано на рис. 1, Z-Wave складається з контролера, який приймає команди від користувача через мобільні телефони; тоді кожен вузол отримує команди від контролера Z-Wave.

Проаналізовано загрози для розумних будинків за допомогою протоколу Z-Wave та проведено експерименти з урахуванням сценаріїв атак. За допомогою цих модельованих атак та аналізу є намір зменшити ризики пристроїв, які використовуватимуть протокол Z-Wave. Для цього було застосовано наступний метод. Детально проаналізований процес зв'язку та процес обміну ключами між контролером та вузлами. За допомогою цього процесу були досліджені проблеми, що виникли під час початкового обміну мережевими ключами між контролером та вузлом. Проаналізована детальна схема потоку даних (DFD) для пристроїв Z-Wave, реалізованих у розумних будинках, яка ілюструвала типи даних, якими обмінюються пристрої в системі Z-Wave. Систематично ідентифікувалися загрози за допомогою моделі STRIDE, яка є репрезентативним методом моделі загроз. Виконано три можливі сценарії атаки. Експерименти з використанням комбінації сценаріїв, що

піддаються атаці, продемонстрували, що мешканці розумних будинків, які використовують пристрої Z-Wave, можуть отримати серйозні пошкодження.

Проаналізовано середовище Z-Wave, яке використовується в розумних будинках із використанням DFD рівня 2; репрезентативний метод моделювання загроз отримав 46 загроз через STRIDE. Виявлено та протестовано три вектори атак, тобто відмова в обслуговуванні (DoS), прошивка в ефірі (FOTA) та віддалена атака управління режимом додаткового режиму на протокол Z-Wave. Аналіз протоколу Z-Wave разом з експериментами на пристроях Z-Wave в розумному будинку продемонстрував, що сучасні розумні будинки є вразливими.

2.1.1 Протокол Z-Wave

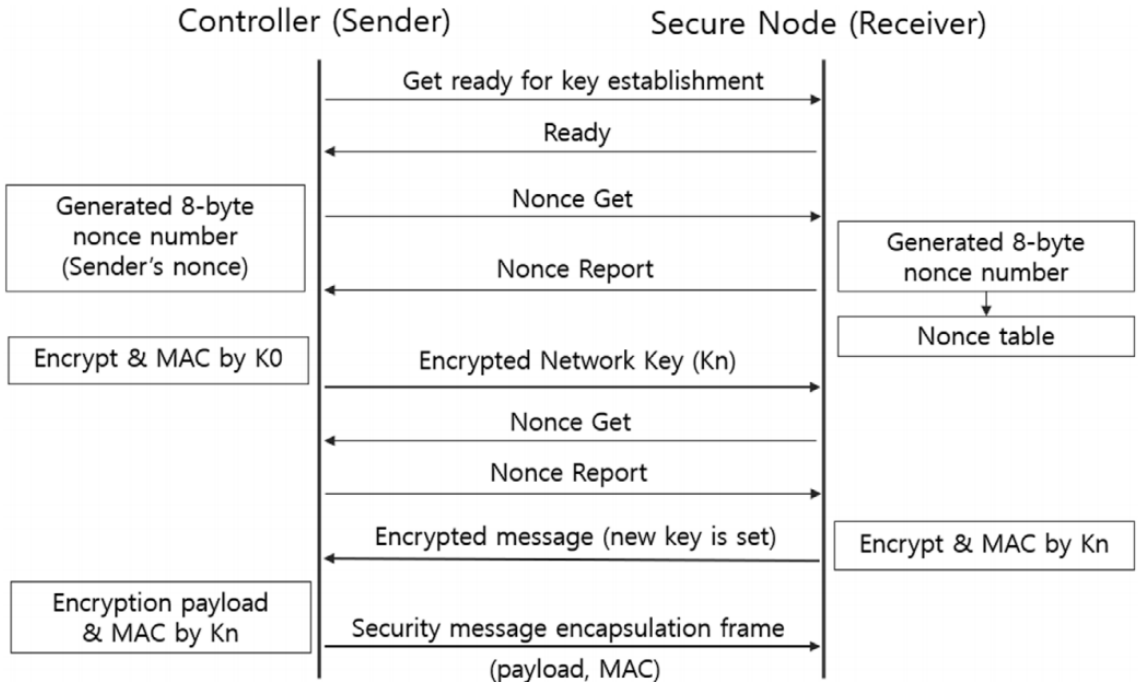


Рисунок 2.2 – Мережевий зв'язок Z-Wave

Протокол Z-Wave складається з чотирьох рівнів:

1. Рівень додатків відповідає за контроль корисного навантаження в кадрах, що приймаються або передаються;
2. Мережевий рівень відповідає за маршрутизацію кадрів, сканування топології та оновлення таблиці маршрутизації;

3. Рівень MAC / рівень передачі, керує передачею даних між двома вузлами і відповідає за підтвердження кадру, перевірку даних та повторну передачу, щоб забезпечити що очікувані вхідні передачі залишаються активними;
4. Фізичний рівень має функцію багаторазового доступу з сенсом несучої, щоб уникнути зіткнення під час бездротового зв'язку. На фізичному рівні використовується модуляція та присвоєння каналу радіочастот (РЧ), додавання преамбули на передавачі та синхронізація на приймачі за допомогою преамбули. ITU-T G.9959 визначає фізичний та MAC-рівні для вузькосмугових прийомопередавачів цифрового радіозв'язку короткого діапазону. Усі виробники дотримуються специфікацій PHY / MAC для забезпечення сумісності [2].

Z-Wave вперше був представлений в незахищеному режимі, і його постійно вдосконалювали для забезпечення безпеки. У незахищеному режимі дані передаються у вигляді простого тексту, і автентифікація між кожним вузлом не потрібна. Таким чином, для вирішення цієї проблеми був розроблений режим командного класу безпеки Z-Wave (S0). Z-Wave S0 забезпечує шифрування та автентифікацію повідомлень. Мережевий зв’язок між контролером (відправником) та вузлом (одержувачем) у режимі S0 проілюстровано на рис. 2.2.

2.1.2 Z-Wave кадру

Кадр Z-Wave містить заголовок, дані програми та циклічну перевірку надмірності (CRC), як показано на рис. 2.3. Заголовок складається з:

– домашнього ідентифікатора для ідентифікації персональної мережі (PAN) контролера;
– вихідного ідентифікатор вузла, щоб ідентифікувати вузол відправника пакетів;
– властивості 1 і 2, які мають кілька застосувань;
– довжина, яка є загальною довжиною кадру;
– порядковий номер, для забезпечення надійності зв'язку;
– ідентифікатор вузла призначення для ідентифікації отриманого вузла.

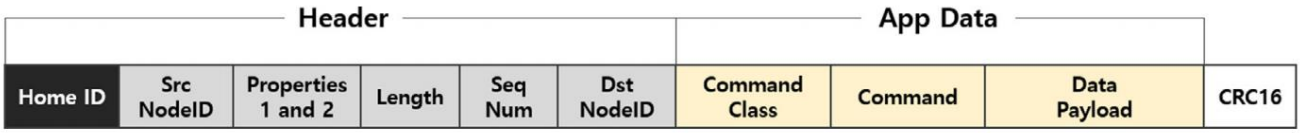


Рисунок 2.3 – Формат кадру Z-Wave

Дані програми Z-Wave містять три елементи, а саме:

1. Клас команд, для ідентифікації конкретних наборів команд;
2. Команда для ідентифікації конкретної команди;
3. Корисне навантаження даних, що є параметрами, які використовується в командах.

2.1.3 Механізм шифрування та автентифікації

У режимі S0 на початковому кроці мережевий ключ (Kn) обмінюється між контролером та вузлом. Він генерується за допомогою апаратного генератора псевдовипадкових чисел (PRNG) разом із наявним тимчасовим ключем за замовчуванням в контролері Z-Wave, як показано на рис. 5. Створений мережевий ключ зберігається в енергонезалежному довільному доступі пам'яті (NVRAM). Він ніколи не використовується безпосередньо для шифрування або надсилання повідомлення про автентифікацію. Після генерації мережевого ключа, він надсилається на вузол Z-Wave на етапі обміну ключами, як показано на рис. 2.2.

Frame	Element	Description
Header	Home ID	Identify PAN of controller
	Source Node ID	Identify a sender node
	Properties 1 & 2	Several options
	Length	Total frame length
	Sequence Number	Ensure reliability of communication
	Dest. Node ID	Identify a receive node
App data	Command class	Identify specific command set
	Command	Identify specific command
	Data payload	Parameters used in command

Рисунок 2.4 - Елементи та опис кадру Z-Wave

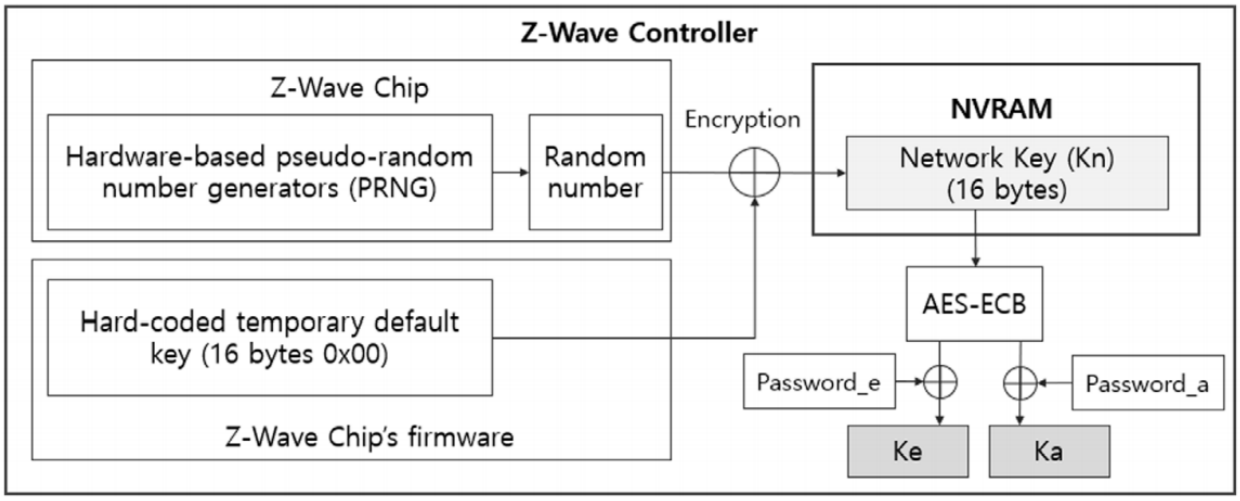


Рисунок 2.5 – Початкове створення ключа

Після успішної відправки мережевого ключа від контролера на вузол, контролер і вузол отримують два нових 16-байтових ключі (K_e та K_a) для шифрування та автентифікації, як показано на формулі 1 для шифрування кадрів даних між контролером та вузлом. K_a - ключ автентифікації джерела даних, який використовується для блокування зловмисних вузлів від підключення до контролера або вузла. K_e генерується для шифрування, як показано в рівнянні. Для автентифікації кожного вузла генерується K_a, як показано в рівнянні. $Password_e$ та $Password_a$ (16 байт 0xAA та 0x55 відповідно) (формула 2) повторюються 16 разів і зберігаються у статичній пам'яті довільного доступу (SRAM).

$$K_e = AES - ECB(K_n, Pass\omega d_e) \quad (1)$$

$$K_a = AES - ECB(K_n, Pass\omega d_a) \quad (2)$$

Після генерації ключа шифрування (K_e) та ключа автентифікації повідомлення (K_a) корисне навантаження, що включає команди, зашифровується, і генерується код автентифікації повідомлення (MAC). Формули для шифрування повідомлення та генерації MAC представлені в рівняннях.

$$C = AES - OFB(K_e, IV \,||P) \quad (3)$$

$$MAC = AES - CBCMAC(K_a||IV||SH||SRC||DST||LEN||C) \quad (4)$$

Як показано на формулі 3, текстове повідомлення (C) шифрується за допомогою методу AES-OFB з вектором ініціалізації (IV) та корисним навантаженням як вхідними значенням; MAC генерується методом AES CBC-MAC з K_a, IV (nonce

відправника + nonce одержувача, 16 байт), SH (заголовок послідовності, 1 байт), SRC (ідентифікатор джерела, 1 байт), DST (ідентифікатор призначення, 1 байт), LEN (довжина корисного навантаження) та C (текстове повідомлення шифру) як вхідні значення. Після створення тексту шифру (C), створюється MAC за допомогою рівнянь, зашифрований фрейм команд для управління іншим вузлом закінчено. Зашифровані та передані пакети показані на формулі 4. Пакети організовані відповідно кольорам.

Доставлені пакетні дані в реальних мережах, як показано на формулі 4, можна порівняти з форматом кадру, що описані на рис. 2.3. Кожен байт інформації включає:

- F9 A6 C0 C8 (домашній ідентифікатор);
- 01 (ідентифікатор вузла джерела);
- 81 00 (властивості 1 та 2);
- 23 (довжина);
- E5 (порядковий номер);
- 1D (ідентифікатор вузла призначення);
- 98 81 (клас команди);
- FC A2 8B 0A 6E 38 73 1F (IV), F6 F4 A7 1A (зашифровані команди даних та корисне навантаження даних);
- D2 (ідентифікатор відсутності приймача);
- 44 69 4A F8 AD 28 D1 0C (код автентифікації повідомлень);
- DE CF (CRC, 16 байт).

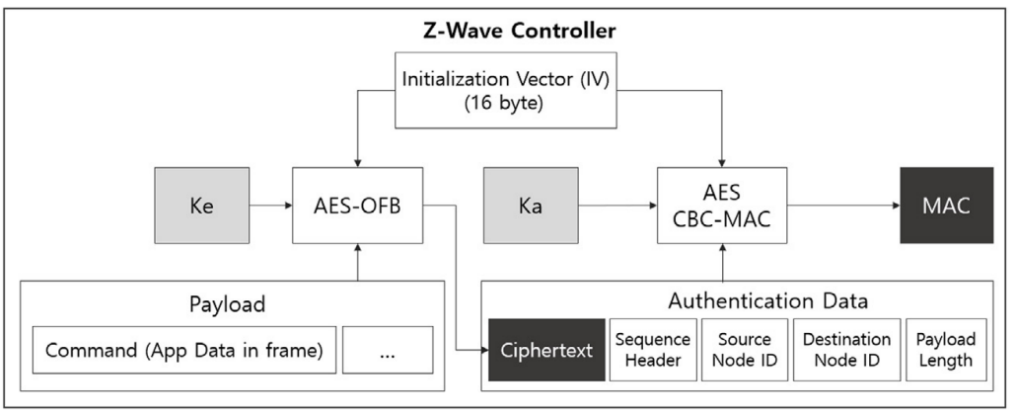


Рисунок 2.6 – Шифрування корисного навантаження та генерація MAC

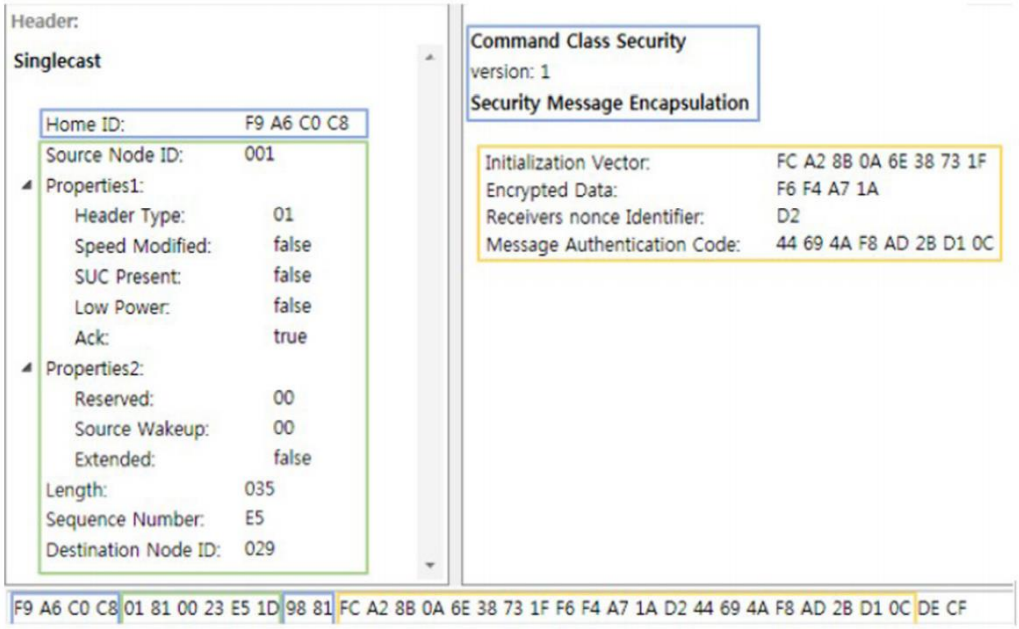


Рисунок 2.7 – Інформація про кадр Z-Wave S0

2.1.4 Спільні ключові проблеми мережі

У Z-Wave S0 мережевий ключ (K_n) спочатку обмінюється між контролером і вузлом для генерації ключів для шифрування (K_e) та автентифікації повідомлень (K_a). На початковому кроці трьома значеннями, створеними для обміну ключами, є мережевий ключ, $Password_e$ та $Password_a$. До обміну ключами мережевий ключ дорівнює 16 байт 0x00, і коли початковий обмін ключами завершується, генерується конкретне значення, як описано на рис. 2.8.

Однак, оскільки жорстко закодовані $Password_e$, $Password_a$ та мережевий ключ можуть бути відомі через перехоплення під час початкового обміну ключами, цей метод здійсненний, якщо перехоплення виконується на початковому етапі процесу обміну ключами. Отже, зловмисник може розшифрувати зашифрований кадр за допомогою перехопленного мережевого ключа. Наприклад, на рис. 2.9 показано, що зашифрований зразок даних 0x1E92A4 розшифровано до 0x009807. Зловмисник може додати 0x00 на початку даних, щоб створити зашифрований пакет.

Name	Before key exchange	After key exchange
Network key (K_n)	16-byte 0x00	446FBA73B03AD7456920F8D80BC1275D
$Password_e$	16-byte 0xAA	16-byte 0xAA
$Password_a$	16-byte 0x55	16-byte 0x55

Рисунок 2.8 – Дані ключа Z-Wave з жорстким кодуванням даних у S0 пристрої

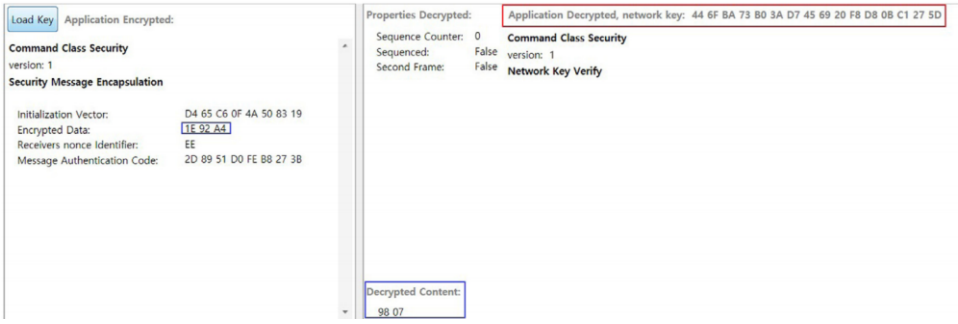


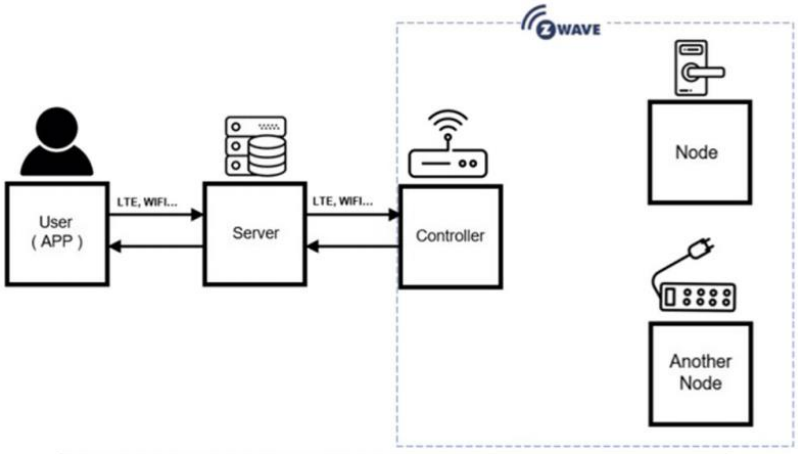
Рисунок 2.9 – Розшифровані дані S0 із зашифрованих даних за допомогою ключа (K_n)

2.2 Методологія

Методологія включала чотири етапи. Підготувати тестовий стенд для аналізу вразливостей пристроїв Z-Wave. Пояснити роботу методу DFD у мережі Z-Wave в реальному середовищі. Прийняти модель загроз STRIDE для виявлення загроз та вразливостей. Створено дерева атак для сценаріїв атак Z-Wave та Z Wave.

2.2.1 Підготовка пробного стенду

На рис. 2.10 показано положення випробувального стенду та опис кожного об'єкта. Пристрої Z-Wave, що використовуються в цьому дослідженні, дотримуються частоти 923,10 МГц. Випробувальний стенд був побудований на основі «Z-Wave IoT Home Service».



Components	Description
User(App)	Send command to Server by App
Server	Send command to Controller
Controller	Controls multiple nodes with Z-Wave
Node	Door lock controlled by Controller
Another node	Multi-tab controlled by Controller

Рисунок 2.10 – Навколишнє середовище та опис Z-Wave

2.2.2 Аналіз діаграми потоків даних

DFD - це метод представлення або виведення потоку даних у процесі або системі. DFD корисні для виявлення вразливостей продукту. Багато дослідників безпеки використовують DFD для системного визначення вразливості цільових продуктів. Для генерації DFD спочатку створюється контекстна діаграма. Контекстні діаграми - це найбільш абстрактні діаграми рівня цільової екосистеми. Більш високий рівень (рівень 0) DFD означає, що потік даних є більш абстрактним, тоді як нижчий рівень DFD (рівень 2) вказує на те, що потік даних є більш конкретним. Більшість досліджень, пов'язаних з DFD, аналізували діаграми рівня 0 або 1 рівня. У роботі проаналізувано DFD рівня 1 і рівня 2 для реальних пристроїв Z-Wave в екосистемі розумного будинку, як показано на рис. 2.11 та у додатку А відповідно. У діаграмі коло представляє процес. Він обробляє отримані дані відповідно до заздалегідь визначеної процедури та видає дані нового типу. Квадрат позначає сутність, тобто абстрактне поняття, де починається потік даних. Вузли та контролер можуть бути включені як сутність. Стрілка вказує на потік даних. Довірена межа - це роздільна лінія, яка вказує область, в якій змінюється рівень повноважень даних. Зберігання даних зберігає вхідні дані та видає збережені дані на запит процесу.

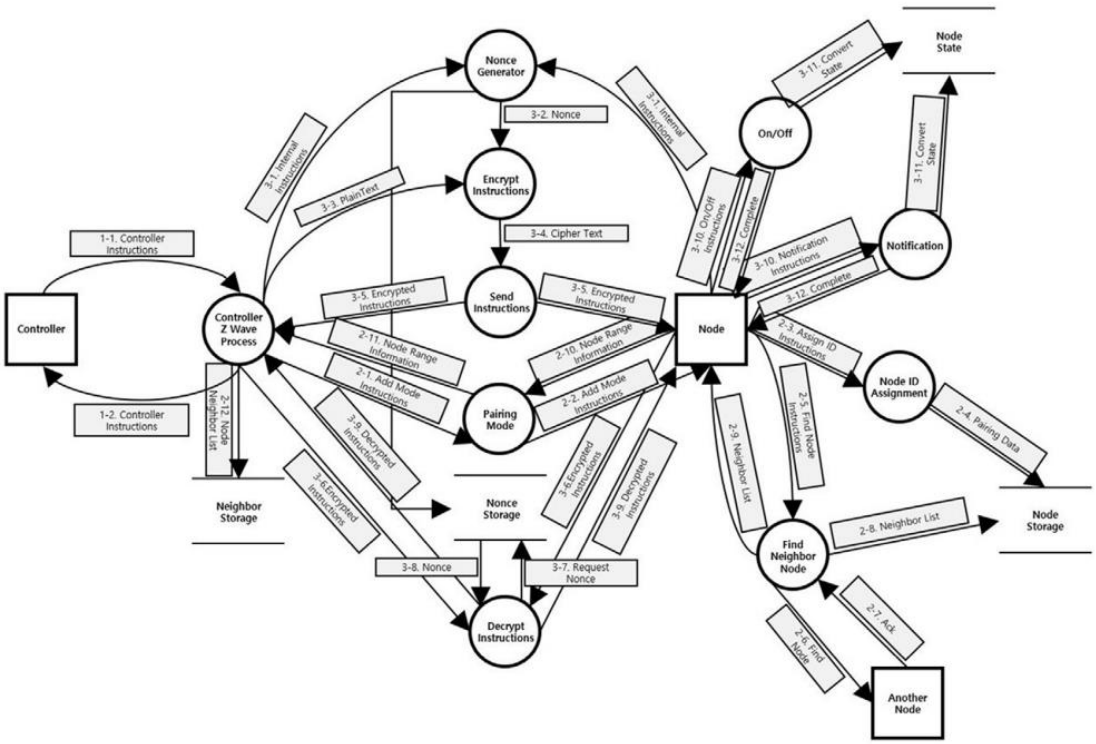


Рисунок 2.11 – Діаграма протоколу Z-Wave рівень 1

2.2.3 Визначення моделі Z-Wave STRIDE

Виведено загрози на основі довіреної межі, потоку даних та процесу, зображеного в DFD рівня 2 Z-Wave. Прийнято модель STRIDE, надану корпорацією Майкрософт, яка є добре відомим методом моделювання загроз для систематичного виявлення загроз безпеці на пристроях [3]. Методи STRIDE ідентифікують підробку, фальсифікацію, відмову, розголошення інформації, DoS та підвищення погроз безпеці, які можуть виникати внаслідок автентичності, цілісності, недоторканості, конфіденційності, доступності та авторизації даних. Підробка відноситься до загрози шахрайства з особистими даними. *Атаки підміни* можна уникнути шляхом реалізації процесу автентифікації в потоці даних або зовнішньої сутності, наприклад, введення користувачем. *Втручання* - це тип загрози, який модулює дані і може бути пом’якшений за допомогою криптографічного хеш-алгоритму для збереження цілісності даних. Цей тип загрози може виникати в процесі, потоці даних або зберіганні даних. *Відмова* передбачає напад заперечення і може бути уникнена за допомогою рішення проти відмови, такого як електронні підписи. Цей тип загроз може впливати на зовнішні фактори, такі як користувачі, зберігання даних або інші процеси. *Розкриття інформації* відноситься до загрози розкриття інформації і може бути уникнено шляхом покращення конфіденційності за допомогою таких методів, як шифрування. Цей тип загрози може існувати в процесах зберігання даних і потоках даних. *DoS* стосується загрози, яка атакує доступність даних; його можна уникнути за допомогою декількох заходів для забезпечення доступності даних. Це може існувати в процесах, процесах зберігання даних та потоках даних. *Підвищення привілеїв* - це тип загрози, який може виникнути під впливом зовнішніх факторів, таких як користувачі процесів. Як правило, загрози можна визначити двома методами. По-перше, шляхом виведення загрози на основі кожного елементу екосистеми Z-Wave; по-друге, шляхом виявлення цих загроз за трьома критеріями: мережа, хост та програма. Цей метод виводить список загроз на основі різних відомих загроз, які можуть існувати в кожному елементі. У дослідженні використовується перший метод для отримання загроз. Наприклад, загроза DoS (D) для компонента 2–2 може виникати в DFD рівня 2. Щоб описати це, можна вивести таку загрозу, як

«Глушення перешкоджає передачі кадру презентації передачі, тим самим порушуючи процес сполучення». Було виявлено всі можливі загрози для кожного компонента та отримано загалом 46 загроз, перелічених у додатку А.

2.2.4 Створення векторів атак

Виведено 46 загроз за допомогою DFD та STRIDE, і відповідно створено дерево атак. Мета дерева атак - отримати повний контроль над розумним будинком. Для цього важливо взяти під контроль мобільний телефон користувача, контролер та кожен вузол мережі Z-Wave. У дослідженні буде зосереджено увагу на вразливостях протоколу Z-Wave, було виключено атаку на доступ до мобільного телефону користувача. Виведено п'ять векторів атак для протоколу Z-Wave для досягнення кінцевої мети, тобто повного контролю над розумним будинком. П'ять векторів атак - це атаки відтворення Z-Wave, пульт дистанційного керування Z-Wave, DoS-атаки Z-Wave, атаки Z-Wave FOTA та дистанційне управління додатковим режимом. Перевірено три вектори атак Z-Wave, крім атаки відтворення Z-Wave та атаки дистанційного керування, та пояснено деталі кожного нападу. Рис. 12 ілюструє три основні тестові шари для атаки Z-Wave для оцінки вразливостей Z-Wave S0. Крім того, описи та передумови трьох векторів атак наведені на рис. 2.12.

Attack vector	Description	Prerequisites
(i) Z-Wave DoS	With specific crafted data, it is possible to disable the features of Z-Wave IoT devices.	The attacker must be in the same network area as the controller.
(ii) Z-Wave FOTA	Vulnerabilities are identified when the Z-Wave firmware is updated via wireless connection.	The attacker must know the network key value by snipping during the pairing process.
(iii) Z-Wave remote add-mode control	With remote add-mode vulnerabilities, it is possible to control any Z-Wave device remotely.	The attacker must know the admin ID and password of the router connected the controller.

Рисунок 2.12 – Опис та передумови трьох векторів атаки

2.2.4.1 Z-Wave DoS

Під час аналізу процесу зв'язку Z-Wave виявлено, що довгі кадри Z-Wave діляться та передаються через кілька невеликих кадрів. Таким чином, ми припустили, що на Z-Wave може бути здійснена атака пінг-смерті. Z-Wave використовує командний клас, який називається транспортною службою, для розділення довгих пакетів.

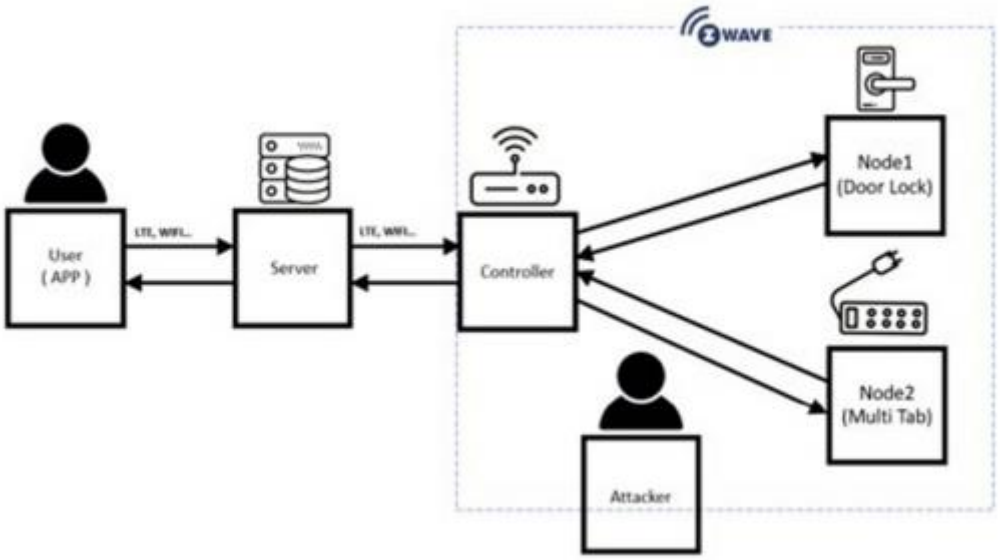


Рисунок 2.13 – Z-Wave DoS

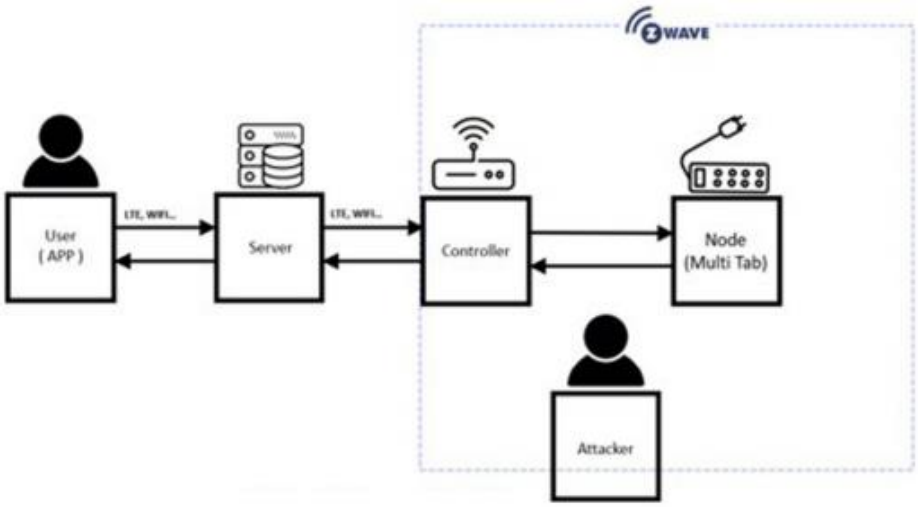


Рисунок 2.14 – Z-Wave FOTA

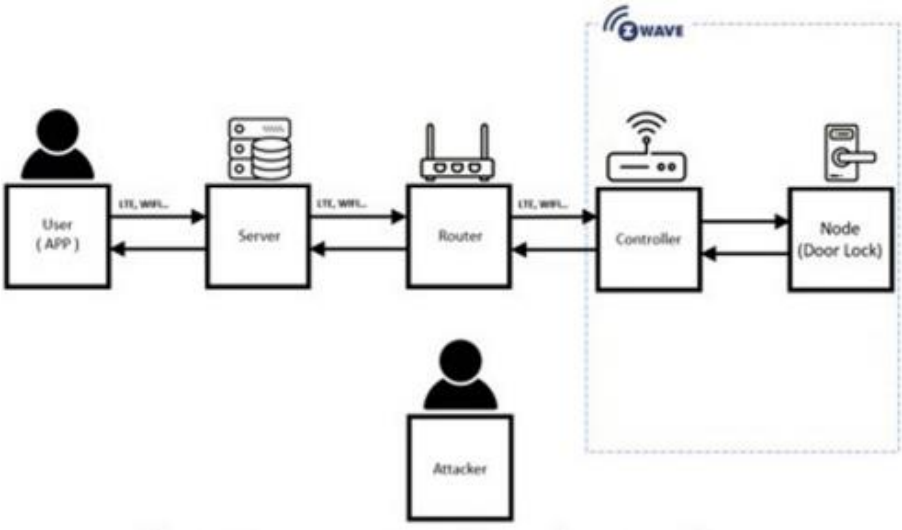


Рисунок 2.15 – Віддалене управління додатковим режимом Z-Wave

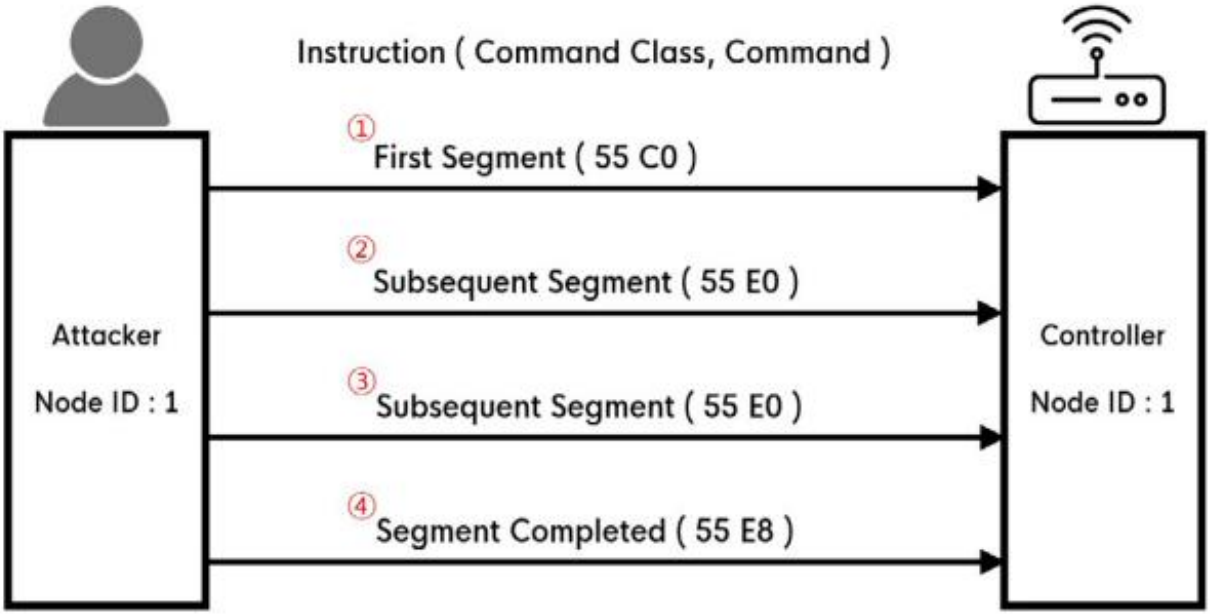


Рисунок 2.16 – Метод сегментації кадру Z-Wave

Процес використання команди транспортної послуги є таким:

1. «Перший сегмент» довгого пакету, який приймається контролером, становить (0x55 C0);
2. Контролер отримує два «наступні сегменти» (0x55 E0);
3. Контролер отримує значення (0x55 E8) як останній отриманий пакет «закінчений сегмент», як показано на рис. 2.16.

Усі кадри, які розділені після першого сегмента, обробляються як наступні сегменти. У цьому випадку, якщо зловмисник надсилає закінчений сегмент пакета після поділу кадру, приймаючий вузол виконує звичайні команди управління після процесу злиття. Підготовлене середовище моделювання для DoS-атак, як показано на рис. 2.13. Для спроби DoS-атаки розроблено «WYP Z-Wave Spoofing Tool», як показано на рис. 2.17, і підготувлено два вузли (дверний замок та мульти-вкладинку), які були підключені до контролера. Було розширено кадр nonce get і він постійно надсилався до контролеру.

Кадр nonce get був розділений і надісланий, оскільки у версії безпеки S0, коли контролер надсилає кадр nonce get на вузол, вузол відповідає на нього у звіті nonce. Процес атаки Z-Wave DoS включав наступні кроки, як показано на рис. 2.18:

Рисунок 2.17 – Z-Wave Spoofing Tool

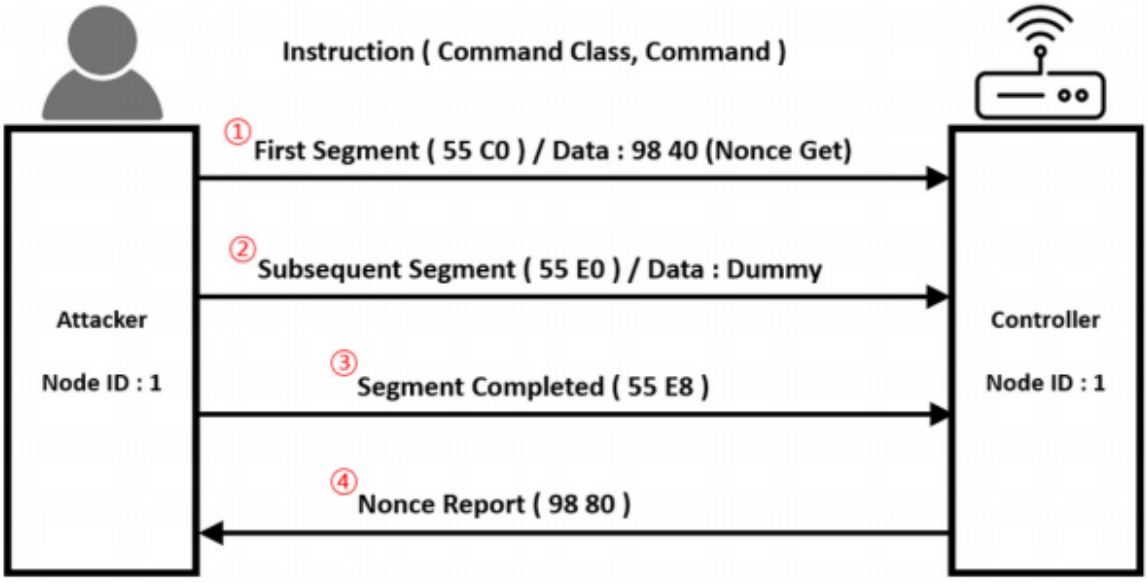


Рисунок 2.18 – Z-Wave процес атаки DoS

1. Контролеру було надіслано значення (55 C0) як перший сегмент із значенням даних 98 40 (nonce get);
2. Контролеру надіслано значення 55 E0 як наступний сегмент із фіктивним значенням даних;

3. Контролеру надіслано 55 E8 як завершену сегментну ціль;
4. Контролер надіслав 98 80 зловмиснику у вигляді звіту про відсутність. Тоді було надіслано довгі дані у вигляді численних значень (0x11) після 0x98 40 для виконання DoS-атаки. Нарешті, ми виявили, що вузол 1 (дверний замок) і вузол 2 (мульти-вкладинка) не можуть підключитися до контролера.

2.2.4.2 Z-Wave FOTA

FOTA вказує на те, що мікропрограма оновлюється бездротовим способом. Z-Wave також включає функцію FOTA, яка вимагає необхідної інформації перед оновленням прошивки. Процес FOTA продемонстровано на рис. 2.18. Коли видається інформація про запит від вузла для оновлення за допомогою прошивки Md Get, отримується звіт про прошивку Md. Звіт про прошивку Md складається з шести елементів: ідентифікатор виробника, ідентифікатор прошивки, контрольна сума прошивки, оновлення мікропрограми, кількість цільових програм та максимальний розмір фрагментів. Запит на оновлення мікропрограми Md Get frame повідомляє, що мікропрограма буде оновлена відповідно до інформації, отриманої від оновлення мікропрограми Md request get. Якщо отримана інформація про мікропрограму збігається із запитаною інформацією про мікропрограму, мікропрограма оновлює звіт Md, який надсилається для сповіщення про оновлення. Таким чином, дані прошивки сегментуються та передаються. Було підтверджено, що деякі продукти S0 реагували на незахищений FOTA. Було припущено, що можливо вкрасти мережевий ключ, розмістивши шкідливе програмне забезпечення до завершеного сполучення у продукті S0. Для цього використано функцію оновлення мікропрограми контролера ПК для виконання оновлення; однак не було можливості активувати завантажене програмне забезпечення, оскільки воно перевіряє правильне значення автентифікації даних мікропрограми, коли остаточно завантажується з пристрою. Щоб усунути це обмеження, була змінена інформація про код за допомогою контролера ПК.

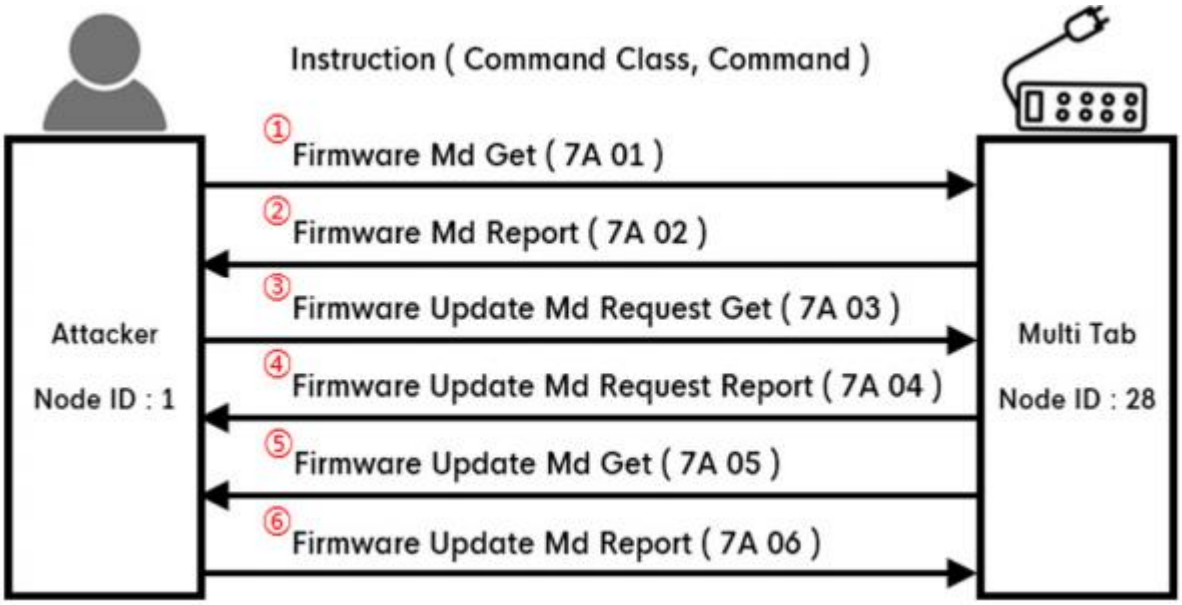


Рисунок 2.19 – Зв'язок Z-Wave FOTA між зловмисником та Multi Tab

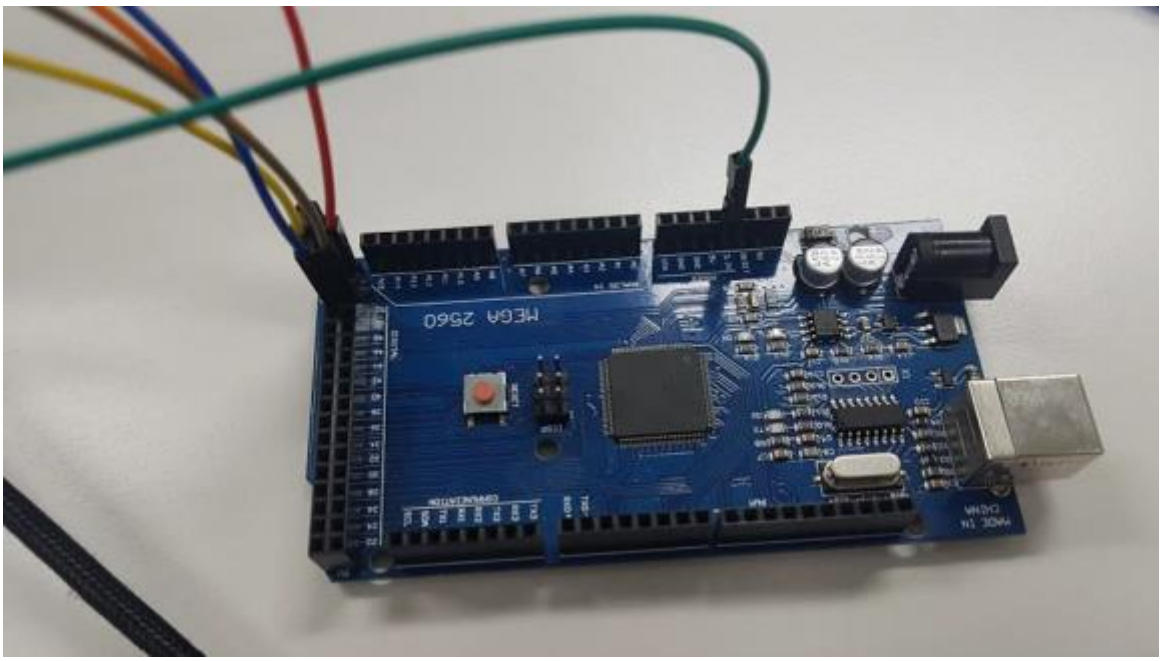


Рисунок 2.20 – Обладнання ATmega2560 для тестування FOTA

Було підтверджено, що атака FOTA можлива за допомогою Z-Wave. Набір інструментів розвитку. Детальні кроки такі:

1. Вихідний код файлу «ZWaveController.sln» у ZIP-файлі контролера ПК, наданий Silicon Labs, був проаналізований та налаштований за бажанням;
2. Під час аналізу функції FOTA виявлено, що важливо отримати прошивку, щоб змінити значення ідентифікатора вузла призначення;

3. Було змінено значення ідентифікатора вузла у вихідному коді «BasicControllerModel.cs»;
4. Після отримання інформації про вбудоване програмне забезпечення від вузла, значення «Ідентифікатор вузла Dst» може бути змінено під час оновлення шляхом модифікації коду;
5. Зміна джерела «BasicControllerModel.cs» дозволило оновити потрібний вузол;
6. Після встановлення точки зупинки в ідентифікаторі перевіряючого вузла можна змінити ідентифікатор вузла та виконати FOTA на потрібному вузлі за допомогою контролера ПК.

Було вирішено провести експерименти на фактичному обладнанні, яке вимагало прошивки. Для вилучення прошивки тестового обладнання використовувалася флеш-пам’ять (M25PE20 та M25PE10). В модулі зв'язку Z-Wave в смарт-замку дверей не було вилучено універсальних асинхронних приймачів-передавачів або портів спільних тестових дій. Тому була зроблена спроба витягти прошивку, надсилаючи команди безпосередньо у флеш-пам’ять через послідовний периферійний інтерфейс зв'язку. Вилучення флеш-пам'яті за допомогою Raspberry Pi та flashrom було першим методом вилучення вбудованого програмного забезпечення, що використовується. Хоча флеш-пам'ять M25PE20, яка намагалася витягти прошивку, була впізнавана програмою flashrom, вона не підтримувала керування на основі команд. Після вдалого витягнення прошивки за допомогою Arduino Mega (ATmega2560), як показано на рис. 2.20, у витягнутій прошивці перші чотири байти містили збережений домашній ідентифікатор, а значення мережевого ключа зберігалося посередині. Було створено двійкові файли, перетворивши шістнадцяткові дані, отримані з флеш-пам'яті. Був використаний інструмент HeX для аналізу таких даних вручну. Домашній ідентифікатор та мережевий ключ були ідентифіковані у простому тексті, а аналіз підтвердив, що дані існували ітеративно з 12-байтним значенням ідентифікації даних. Було завантажено двійковий файл із підробленим підписом і підтверджено, що він працює стабільно залежно від пристрою.

2.2.4.3 Дистанційне керування режимом додавання

Режим додавання - це режим роботи, який повинні виконувати пристрої Z-Wave, щоб з'єднуватися між собою. Щоб контролер і вузол почали сполучення, обидва пристрої повинні бути в режимі додавання. Як правило, більшість вузлів можуть увійти в режим додавання, коли кнопка на вузлі натискається більше 3 секунд. Однак у випадку з контролером, для зручності користувача, режим контролера можна змінити віддалено за допомогою програми користувача, не натискаючи жодної кнопки. «Атака віддаленого керування режимом додавання» - це метод атаки, при якому зловмисник примусово перемикає контролер у режим додавання (режим сполучення) ззовні і перехоплює мережевий ключ жертви, поєднуючи з ним вузол зловмисника.

Було підготовлено тестовий стенд для сценарію атаки віддаленого управління режимом додавання, як показано на рис. 2.14. У цьому сценарії атаки передбачалося, що маршрутизатор жертви розташований між сервером та контролером жертви, а зловмисник знає ідентифікатор адміністратора та пароль маршрутизатора. Коли зловмисник успішно отримує доступ до маршрутизатора жертви та виконує атаку MITM, пакет між сервером та контролером може бути проаналізований. Тому зловмиснику легко перемкнути контролер у режим додавання, надіславши аналізовану команду перемикання режиму додавання до контролера.

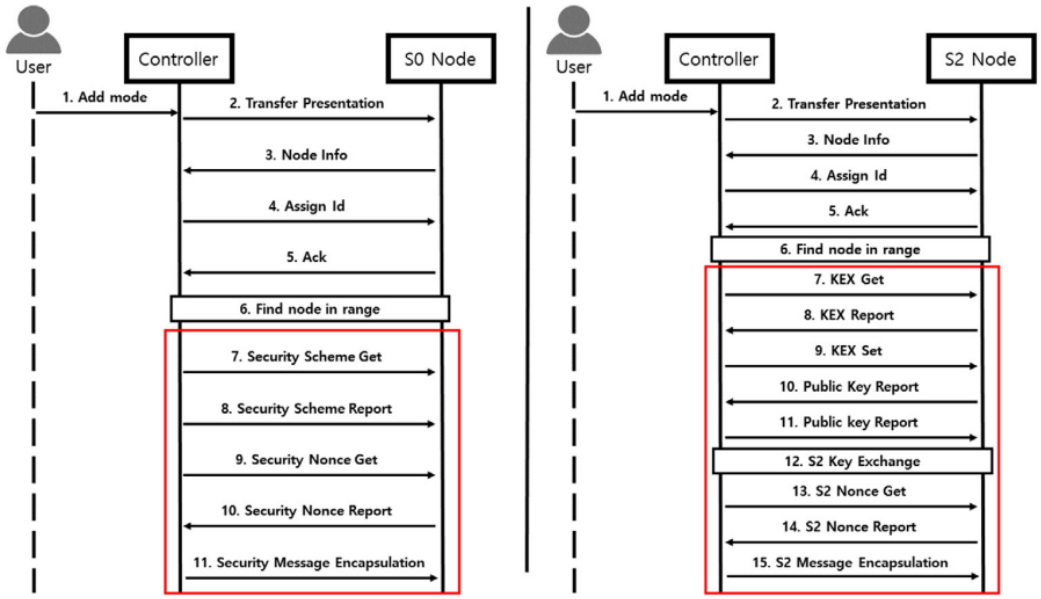


Рисунок 2.21 – Відмінності зв'язку протоколу S0 та S2

Після успішної відправки пакету додаткового режиму, згенерованого зловмисником, можна помітити, що контролер, як правило, переходить у режим додавання та надсилає кадр Z-Wave для створення пари. Було вилучено мережевий ключ.

2.2.4.4 Аналіз Z-Wave S2

Було проаналізовано протокол зв'язку класу команд безпеки та вузлів класу безпеки 2 (S2). Рис. 2.21 показує протокольний зв’язок для кожного з вузлів S0 та S2. Найбільша різниця між протоколами S0 і S2 полягає в тому, що протокол S2 обмінюється ключами між контролером і вузлом, коли користувач запитує режим віддаленого додавання.

Виявлено кілька вразливостей Z-Wave і отримано дві загальні вразливості та ризики (CVE) а саме: вразливість MITM (віддалене управління додатковим режимом) та вразливість DoS Z-Wave. Національна база даних про вразливості (NVD) довела, що DoS-атака можлива для пристроїв ZWave, що використовують протокол безпеки S2. Теоретично, атака S2 DoS можлива, оскільки єдина різниця між протоколами S2 і S0 полягає в «методі обміну ключами та механізмі шифрування», отже, атаки DoS однакові.

Таблиця 2.1 - Застосування протоколів S0 та S2 для кожного сценарію.

Сценарії	S0	S2
Сценарій 1: DoS	O	O
Сценарій 2: FOTA	O	*
Сценарій 3: Режим віддаленого додавання	*	*

2.3 Експеримент

2.3.1 Підготовка випробувального стенду

Рис. 2.22 показано тестові стенди для сценаріїв атаки Z-Wave, що включають реальні пристрої. Зловмисник має фізичний доступ до мережі жертви; отже, він може отримати доступ до внутрішньої мережі жертви через Wi-Fi.

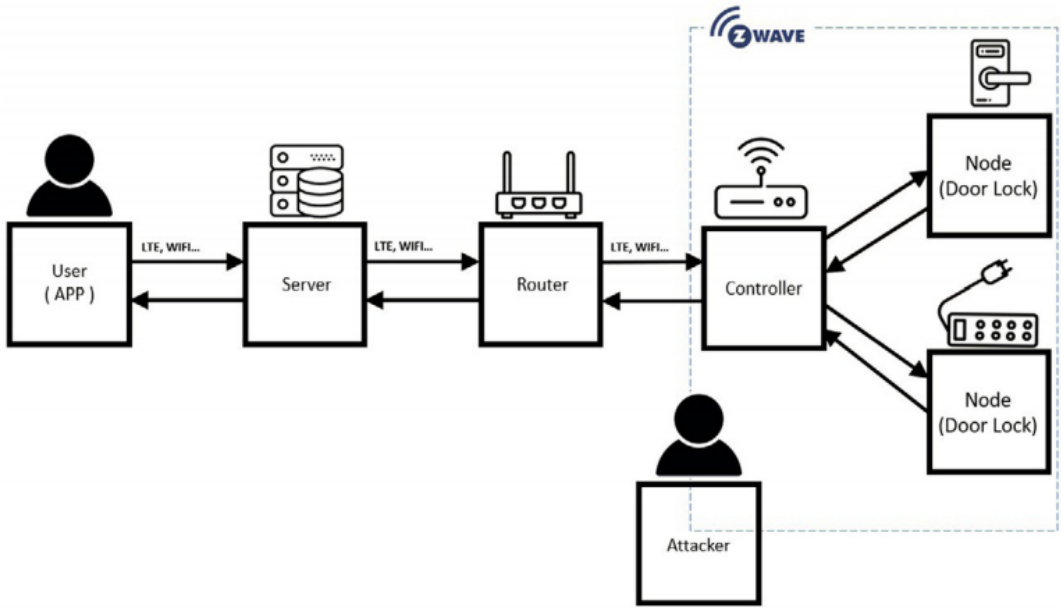


Рисунок 2.22 – Тест сценарію атаки Z-Wave

2.3.2 Результати

Спочатку було здійснено атаку MITM на цільовий маршрутизатор і контролер. Для віддаленої атаки управління додатковим режимом підготовлено проксі-сервер і клієнт, які спілкувались із сервером після захищеної смуги шарів сокета. На рис. 2.15 та 2.23 показані стани до та після того, як зловмисник здійснив атаку віддаленого управління додатковим режимом відповідно.

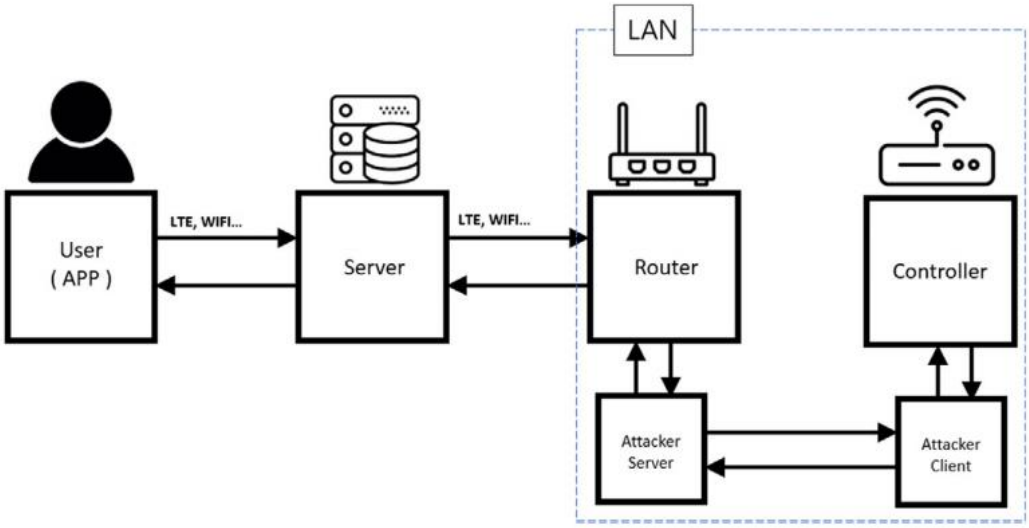


Рисунок 2.23 – Випробувальний стенд після віддаленого управління додатковим режимом

Після успішних атак MITM між маршрутизатором і контролером було зібрано та модифіковано дані між ними. Для передачі даних між пристроєм та центральним контролером будинку обидва повинні мати спільний мережевий ключ, що дозволяє спілкуватися. Коли новий пристрій поєднується через Z-Wave, виконується певний протокол синхронізації для спільного використання мережевого ключа з пристроєм [4]. Щоб зв’язати вузол зловмисника з контролером цілі та надати спільний доступ до мережевого ключа, контролеру було надіслано команду режиму сполучення, ніби ця команда була надіслана із сервера. Рис. 2.21 демонструє стан після успішної атаки в режимі сполучення. Отже, зловмисник отримав мережевий ключ, який використовується для зв'язку між маршрутизатором і контролером.

Було використано інструмент для здійснення віддаленої атаки додаткового режиму. Оскільки мережевий ключ, яким обмінювався цільовий контролер та вузол зловмисника за допомогою сполучення, був ідентичним цільовому мережевому ключу Z Wave, цей мережевий ключ та наш інструмент дозволили зловмиснику керувати цільовими пристроями S0. Крім того, було здійснено DoS-атаки на контролер, щоб запобігти попереджувальній сигналізації користувача, коли пристрій знаходиться під довільним контролем.

Таким чином, навіть якщо зловмисник втрутився посередині, цільовим пристроєм Z-Wave можна керувати, не викликаючи сигналізацію. Експеримент було проведено десять разів, щоб отримати повний доступ до переданого пристрою. Рис. 2.25 показує результати експерименту. Середній час, необхідний тесту для повного контролю, становить 45,291 с.

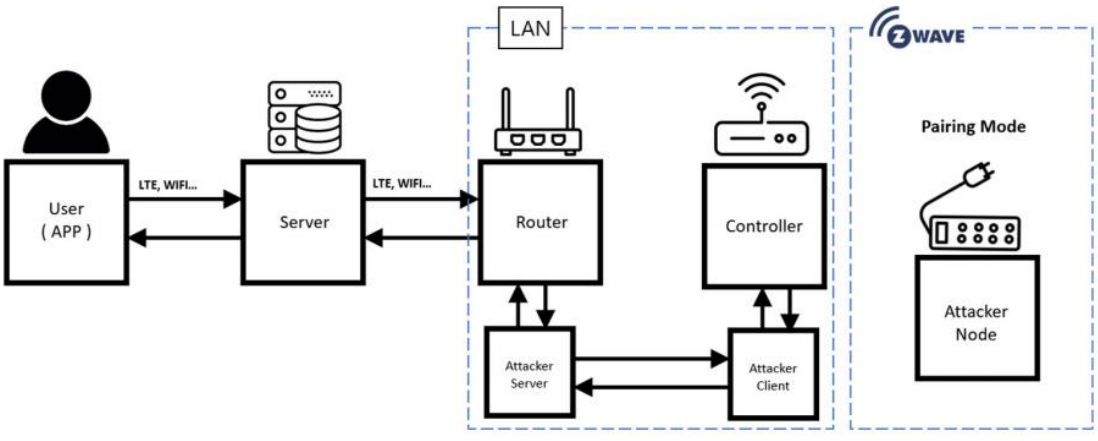
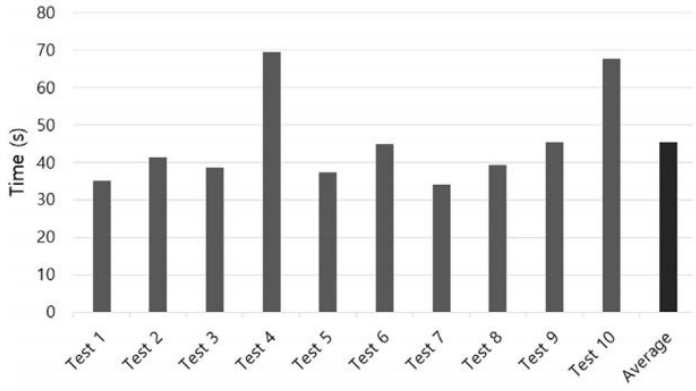


Рисунок 2.24 – Сполучення за допомогою вузла зловмисника



Count	Attack Period		Attack Time (s)
1	17:53:08:045	17:53:43.231	35.186
2	17:56:29.397	17.57:10.676	41.279
3	17:59:52.561	18:00:31.070	38.509
4	18:06:26.219	18:07:35.640	69.421
5	18:09:12.527	18:09:49.885	37.358
6	18:16:04.666	18:16:49.495	44.829
7	18:18:13.676	18:18.47.679	34.003
8	18:19.44.485	18:20.23.919	39.434
9	18:21.11.753	18:21.57.153	45.4
10	18:27.29.889	18:28.37.382	67.493

Рисунок 2.25 – Час атаки для отримання повного доступу

Висновоки до розділу

У розділі проведено аналіз реальних пристроїв Z-Wave, використовуючи такі методи моделювання загроз, як модель DFD та STRIDE. Досліджено три вектори атак Z-Wave, включаючи DoS, FOTA та атаки дистанційного керування додатковим режимом. Проведено експерименти для Z-Wave в реальному середовищі, об’єднавши вектори атаки. Виявлено що якщо комбінувати атаки Z-Wave, вони можуть завдати критичної шкоди мешканцям розумних будинків.

Попередньо спільну ключову проблему було вирішено шляхом підвищення рівня безпеки з S0 на S2. Однак, як і раніше існує потенційна проблема, що мережевий ключ, який використовується контролером для управління кількома вузлами, є однаковим симетричним ключем. Для кращої безпеки контролер повинен розподіляти симетричні ключі, що використовуються для управління вузлами по-різному для кожного вузла. Таким чином, навіть якщо зловмисник викраде мережевий ключ, прикріпивши його до вузла, він не зможе керувати іншими вузлами.

Також необхідно запобігти легкому отриманню значення ключа, яке може забезпечити довгостроковий контроль, періодично змінюючи попередньо спільний ключ випадковим чином для підключеного вузла. Крім того, підключений пристрій повинен мати можливість змінити попередньо спільний ключ у певний час, використовуючи FOTA та PRNG для відкритого ключа, щоб переконатися, що зловмисник не може легко отримати значення ключа. Z-Wave не використовує сегментацію кадру в програмах. Отже, якщо пристрій Z-Wave не вимагає цієї функції, його відключення може ефективно запобігти атакам DoS.

РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ АВТОМАТИЗАЦІЇ РОЗУМНОГО БУДИНКУ

3.1 Обгрунтування вибору середовища розробки та опис архітектури

Для створення унікального інтерфейсу з подальшою інтеграцію у систему розумного будинку Z-Wave, використано бібліотеку OpenZWave. OpenZWave - це міжплатформна бібліотека з відкритим кодом, призначена для того, щоб будь-хто міг додати підтримку пристроїв домашньої автоматизації Z-Wave до своїх додатків, не вимагаючи глибоких знань протоколу Z-Wave.

Z-Wave - це власний протокол бездротового зв'язку, що використовує технологію мережевих мереж. Мережева мережа дозволяє пристроям низької потужності спілкуватися на великі відстані та навколо чорних плям, передаючи повідомлення від одного вузла до іншого. Важливо зазначити, що не всі пристрої Z-Wave працюють постійно, особливо ті, що живляться від акумулятора. Ці вузли не можуть брати участь у пересиланні повідомлень через сітку.

Кожен пристрій Z-Wave відомий як "Вузол" у мережі. Мережа Z-Wave може містити до 232 вузлів. Якщо потрібно більше пристроїв, то кілька мереж потрібно налаштувати за допомогою окремих контролерів Z-Wave. OpenZWave підтримує декілька контролерів, але сам по собі не перетворює мережі, дозволяючи пристрою на одному безпосередньо керувати пристроєм на іншому. Цю функціональність повинна надавати програма.

Вузли Z-Wave можна розділити на два типи: контролери та підлеглі. Контролери зазвичай мають форму ручного пульта дистанційного керування або інтерфейсу ПК. Вимикачі, регулятори яскравості, датчики руху тощо - все це підлеглі. Значення класів команд реплікації контролерів та пристроїв

Усі функції Z-Wave доступні через клас Manager. Хоча це не робить найбільш ефективною структурою коду, вона дозволяє бібліотеці обробляти потенційно складні та важкі для налагодження проблеми, такі як багатопоточність та тривалість життя об'єктів за кадром. Тому розробка додатків спрощена і менш схильна до помилок.

Зв'язок між ПК та пристроями в мережі Z-Wave відбувається асинхронно. Деякі пристрої, зокрема датчики руху, більшу частину часу сплять, щоб заощадити заряд батареї, і можуть отримувати команди лише в режимі сплячого режиму. Тому команда змінити значення в пристрої може відбутися не відразу. Це може зайняти кілька секунд або хвилин, а то й взагалі ніколи не прибути. З цієї причини багато методів OpenZWave, навіть ті, що вимагають інформацію, не повертають цю інформацію безпосередньо. Запит буде надіслано в мережу, а відповідь надіслана додатку через деякий час через систему зворотних викликів сповіщень. Обробник сповіщень буде в основі будь-якої програми, що використовує OpenZWave. Саме тут буде повідомлятися вся інформація щодо конфігурацій та станів пристрою.

Мережа Z-Wave - це динамічна сутність. Контролери та пристрої можна додавати або видаляти будь-який час. Після налаштування мережі це, мабуть, траплятиметься не дуже часто, але OpenZWave та будь-яка побудована на ній програма повинні впоратися все одно. Система зворотного виклику повідомлень використовується для інформування заявки про будь-які зміни в структурі мережі.

Основними міркуваннями при розробці програми на основі OpenZWave є:

– Зв'язок із пристроями Z-Wave є асинхронним і не гарантовано надійним. Не покладайтесь на отримання відповіді на будь-який запит.
– Мережа Z-Wave може змінюватися будь-який час. Обробник зворотного виклику сповіщень програми повинен обробляти всі сповіщення, і будь-яке представлення стану мережі Z-Wave, що утримується додатком, має бути змінено, щоб відповідати. Користувацькі інтерфейси повинні будуватися динамічно на основі інформації, повідомленої у зворотних викликах сповіщень, і повинні мати можливість справлятися з додаванням та видаленням пристроїв.

Деякі користувачі матимуть більше одного контролера Z-Wave, щоб дозволити віддалене розташування або обійти обмеження 232 пристроїв. OpenZWave призначений для використання з декількома контролерами, і всі програми повинні бути написані для підтримки цього.

Немає необхідності зберігати стан мережі та відновлювати її під час наступного запуску. OpenZWave робить це автоматично і призначений для подолання будь-яких змін, що відбуваються в мережі, поки програма не працює.

На основі біблотеки OpenZWave було розроблено середовища систем автоматизації розумних будинків, а саме:

- ansible (через Thrift4OZW) – Сценарій домашньої автоматизації для рубіну
- openzwave-control-panel – веб-інтерфейс для управління та моніторингу мереж Z-Wave
- Open Source Automation – Розширювана платформа автоматизації (лише для Windows)
- zVirtualScenes – програмний контролер сцени для пристроїв ZWave
- ago control – пакет автоматизації домашньої автоматизації на основі обміну повідомленнями AMQP. Також підтримує інші протоколи, крім Z-Wave. Легкий дизайн для роботи на вбудованих системах, таких як Raspberry Pi.
- DomotiGa – програмне забезпечення для домашньої автоматизації з відкритим кодом, яке підтримує Zwave серед інших протоколів.
- Homeautomation – веб-програма PHP, яка підтримує ZWave та інші протоколи HA (наприклад, Tellstick)
- Domoticz – Domoticz - це дуже легка система домашньої автоматизації, яка дозволяє контролювати та налаштовувати різні пристрої, такі як: Світло, Вимикачі, різні датчики / лічильники, такі як Температура, Дощ, Вітер, УФ, Електрика, Газ, Вода та багато іншого.
- IOBroker – це інтеграційна платформа для Інтернету речей, орієнтована на автоматизацію будівель, інтелектуальне вимірювання, навколишнє середовище, автоматизацію процесів, візуалізацію та реєстрацію даних.

Отже, переглянувши всі системи автоматизації, вибрано інтеграційну платформу IOBroker. Переваги цієї платформи: дужа комфортна та багатофункціонланьна панель управління. ioBroker є модульним, тобто побудований з безлічі окремих компонентів. Кожен модуль має певне завдання. Тому, щоб відстежувати, ioBroker має центрального координатора для всіх своїх модулів. Цей

координатор є фоновим робочим js-controller. Він відповідає за централізоване керування даними, а також за управління та зв'язок між усіма модулями. Самі модулі називаються Adapter. Адаптери встановлюються користувачем тільки при необхідності. Веб-інтерфейс адміністрування admin сам по собі є адаптером. Адаптер адміністратора або скорочено «admin» - це інтерфейс управління системою ioBroker. Адмін зазвичай викликається з адресою HTTP: // локальний: 8081.

Коли новий адаптер встановлюється разом з адміністратором, файли адаптера спочатку завантажуються з Інтернету і записуються на диск сервера. Якщо адаптер повинен бути запущений, спочатку генерується Instanz адаптера. Кожен екземпляр адаптера може бути індивідуально налаштований, зупинений і запущений незалежно від адміністратора. Тому кожен екземпляр запускається в своєму власному процесі, який взаємодіє в фоновому режимі з js-контролером ioBroker.

В системі Multihost з декількома серверами ioBroker екземпляри адаптерів також можуть бути розподілені на різних серверах. В результаті навантаження може бути розподілена або додаткове обладнання може бути підключено безпосередньо на місці (наприклад, порти введення-виведення, USB).

Зв'язок між адаптерами, JS-контролерами, базами даних і веб-інтерфейсами здійснюється через кілька з'єднань TCP / IP. Обмін даними відбувається в залежності від обраної налаштування або у вигляді простого тексту, або в зашифрованому вигляді.

ioBroker і адаптери в основному написані на мові програмування JavaScript. Для запуску JavaSript вам потрібне відповідне середовище виконання. Тому ioBroker використовує Node.js. Це середовище виконання доступна для різних програмних платформ, таких як Linux, Windows і macOS. Менеджер пакетів JavaScript npm використовується для установки ioBroker і адаптерів.

ioBroker - візуалізації - огляд та порівняння

VIS

VIS - це класичне рішення візуалізації від ioBroker. Інтерфейс користувача може бути розроблений абсолютно вільно за допомогою різних віджетів. На наступному скріншоті показано редактор візуалізації. Представлення спочатку

створюються за допомогою редактора, на якому можна вільно розміщувати окремі віджети. Далі віджет можна змінити за допомогою властивостей праворуч від редактора.

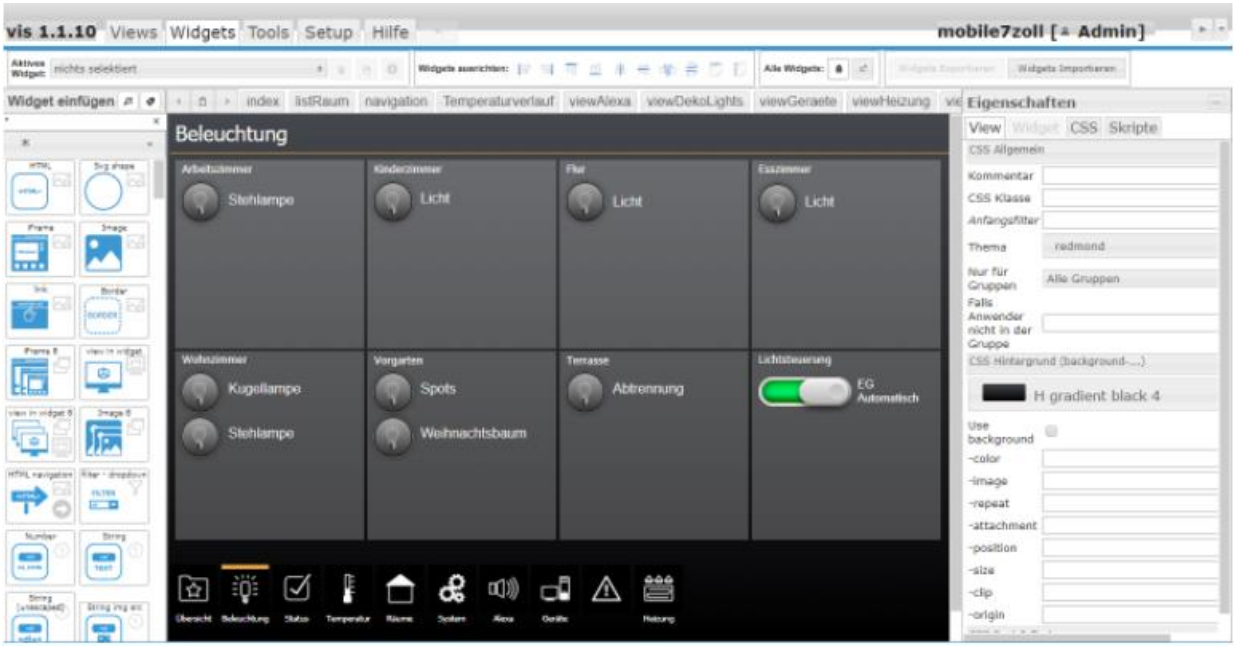


Рисунок 3.1 – VIS візуалізація в ioBroker

За допомогою редактора VIS ви можете створювати повністю мінливі візуалізації розумного будинку, які побудовані та розроблені відповідно до ваших уявлень. Також доступна велика кількість віджетів, які можна розширити за допомогою адаптера. Однак впровадження є складним у багатьох сферах і вимагає багато часу. Крім того, візуалізація адаптована до відповідного кінцевого пристрою, немає адаптивного дизайну для різних кінцевих пристроїв.

2. Material UI

За допомогою Material US можна побудувати динамічну поверхню, компоненти якої, такі як Кімнати або функції можна прочитати з конфігурації ioBroker (списки та об'єкти). Це надає користувальницький інтерфейс, за допомогою якого будинок управління може бути зіставлений без необхідності створювати складний вигляд у VIS.

За допомогою візуалізації інтерфейсу матеріалу ви можете швидко і легко створити сучасну візуалізацію. Списки та метадані об'єктів в ioBroker є основою візуалізації.

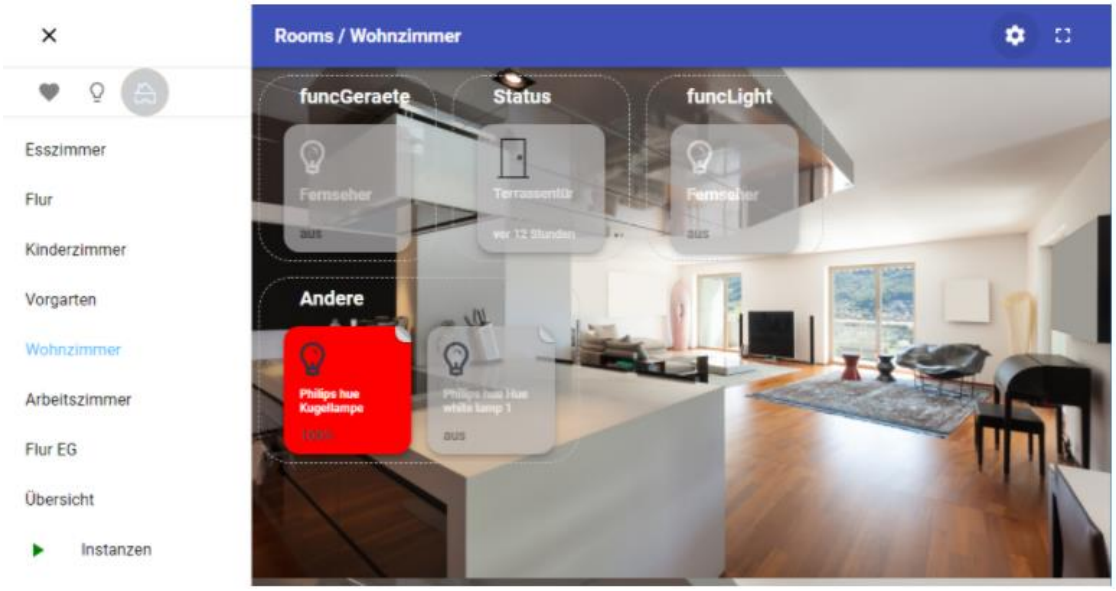


Рисунок 3.2 – Material UI візуалізація в ioBroker

Пускачі та датчики автоматично вставляються у візуалізацію на основі приміщення та торгового призначення (функції). Ви також можете за бажанням додати власні віджети. На момент створення статті адаптер ще тестувався, але він уже відносно великий і може бути використаний для вашої власної візуалізації. Добре доглянута структура об’єкта є основою гарної візуалізації.

3. iQontrol

iQontrol - це суміш Apple Homekit та інтерфейсу Material. Візуалізацію порівняно легко налаштувати за допомогою подань. Потім різні пристрої з точок даних ioBroker можуть бути додані до відповідного виду (наприклад, освітлення).

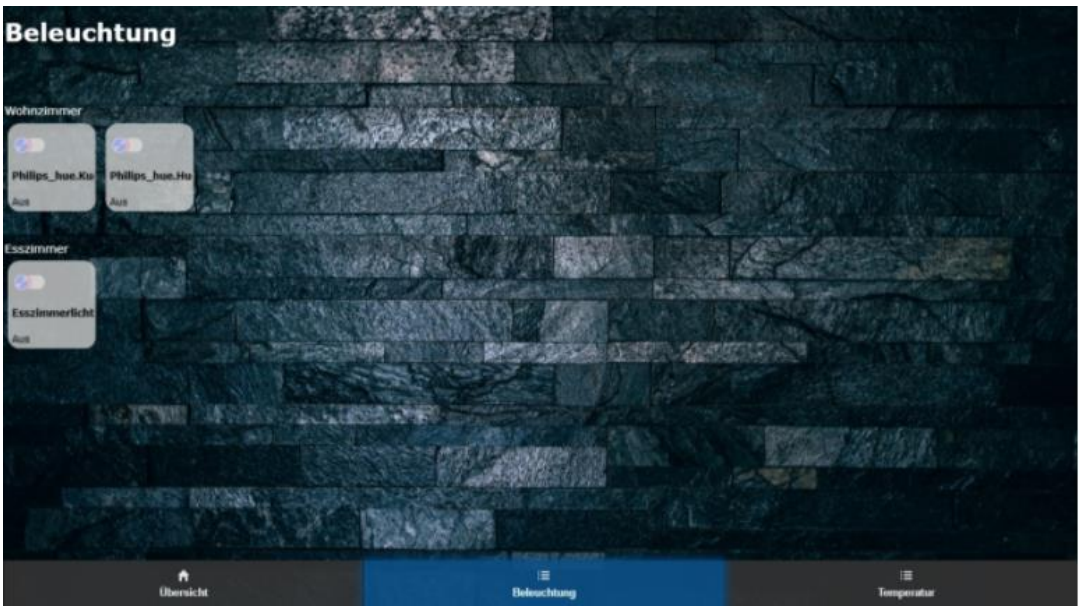


Рисунок 3.3 – iQontrol візуалізація в ioBroker

Я протестував адаптер і побудував частину своєї системи автоматизації будинку приблизно за 1 годину. В основному, мені дуже подобається підхід нового адаптера, я бачу тут великий потенціал. Цей адаптер порівняно простий у розумінні та використанні, особливо для початківців. Адаптер зручний у користуванні та дуже простий, як для початківця створювання унікального візуалзуючого інтерфейсу для користувача.

4. HABPanel

За допомогою HABPanel ви можете швидко і легко налаштувати інтерфейси візуалізації для ioBroker. Візуалізація спочатку походить від openHAB і згодом була доступна для ioBroker. На даний момент адаптер недоступний у виробничому середовищі (станом на серпень 2019 року).

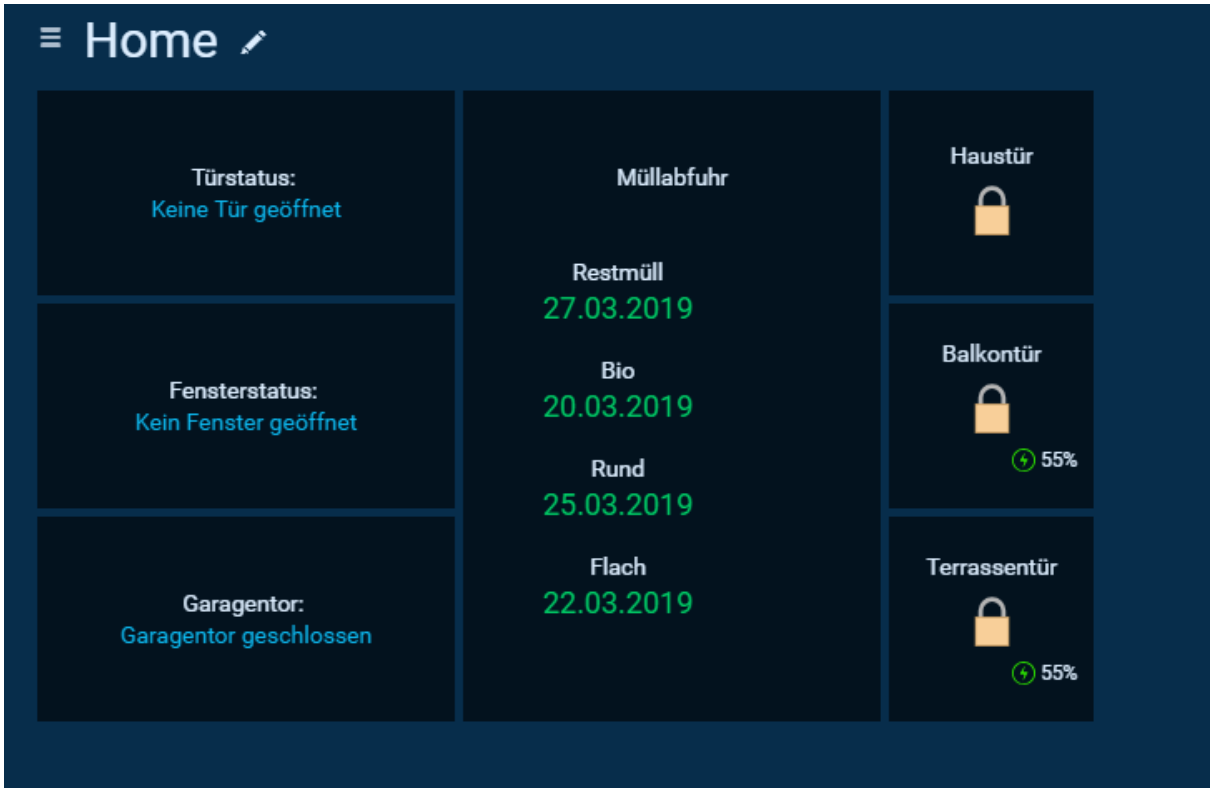


Рисунок 3.4 – HABPanel візуалізація в ioBroker

За допомогою адаптера HABPanel ви також можете швидко реалізувати свою візуалізацію. Мені подобається структура та інтуїтивна робота. Адаптер має велику кількість віджетів, які можна використовувати для візуалізації. Ви також можете використовувати знання HTML, CSS та AngularJS для розробки власних віджетів або імпортування віджетів із спільноти в HABPanel. Я вважаю відсутність адаптивного веб-дизайну недоліком.

5. Lovelace-UI

За допомогою Lovelace ви можете швидко і легко налаштувати інтерфейси візуалізації для ioBroker. Візуалізація спочатку походить з платформи Home Assistant, а згодом також стала доступною для ioBroker. На даний момент адаптер недоступний у виробничому середовищі (станом на серпень 2019 року).

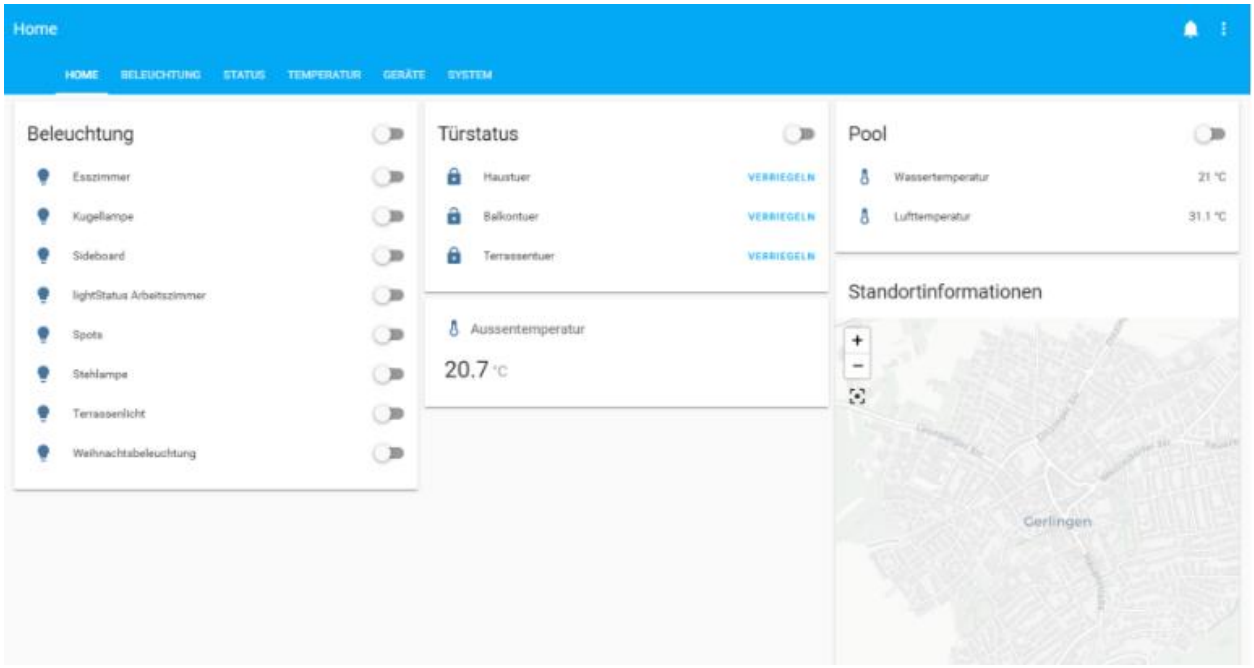


Рисунок 3.5 – Lovelace-UI візуалізація в ioBroker

За допомогою Lovelace ви також можете відносно швидко створювати сучасні візуалізації. Датчики або виконавчі механізми повинні бути активовані в об'єктах для Lovelace. Крім того, розподіл приміщень та функцій також повинен бути доступним. Візуалізація мене абсолютно переконала, оскільки вона має інші додаткові функції, такі як таймери або система сповіщення. Зараз я переношу всю свою візуалізацію на цей новий адаптер.

6. Tileboard

Tileboard - ще один адаптер візуалізації, реалізований від Home-Assistant для ioBroker. На даний момент адаптер недоступний у виробничому середовищі (станом на серпень 2019 року).

Як зазначалося вище, ця форма візуалізації знаходиться на дуже ранній стадії інтеграції в ioBroker. Повна конфігурація адаптера проходить через файл JavaScript, який можна редагувати в конфігурації адаптера. У цей файл можна вставити окремі сторінки, групи та сутності, а також налаштувати окремі сутності.

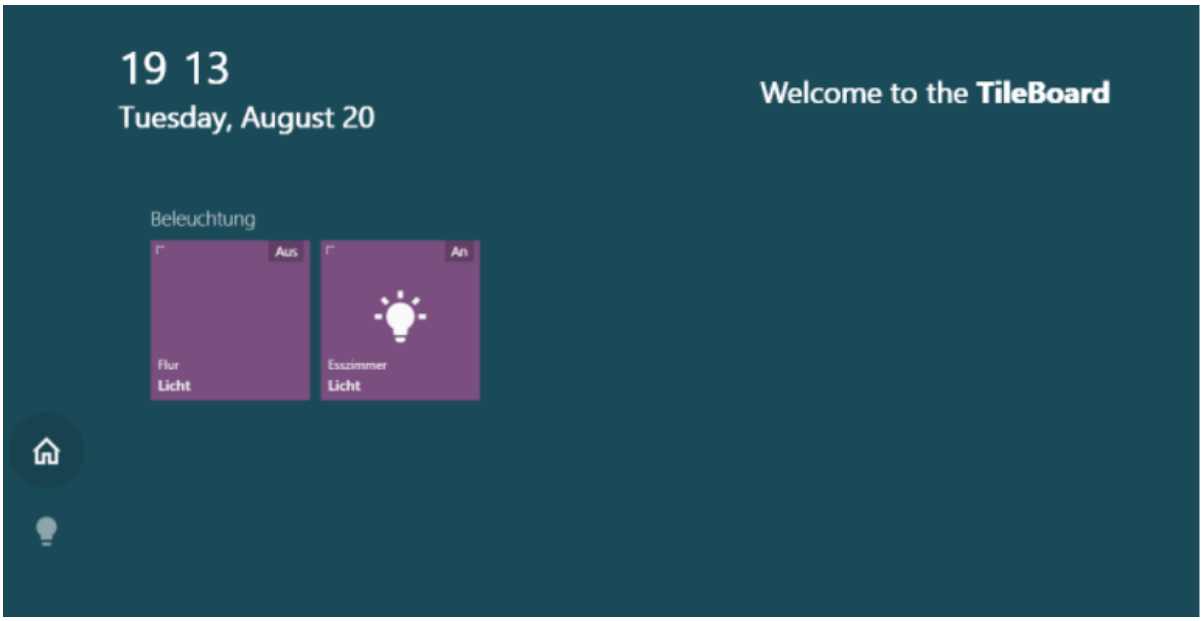


Рисунок 3.6 – Tileboard візуалізація в ioBroker

7. Мобільний

Мобільна візуалізація була першим підходом до динамічної візуалізації. Адаптер не буде надалі розроблятись чи обслуговуватися, тому я не рекомендую використовувати візуалізацію. Я не буду далі описувати або оцінювати це рішення далі.

8. Node-RED візуалізація

Node-RED має вбудовану візуалізацію, яка може бути використана для налаштування власних інформаційних панелей. Основна структура візуалізації відбувається через визначення вкладок та груп, які можна створити безпосередньо в Node-RED. Окремі робочі елементи, напр. для датчиків інформація або лампи потім збираються у вигляді вузлів приладової панелі на одному або декількох потоках. За допомогою цього підходу можна створити цілі інформаційні панелі.

9. Jarvis

Jarvis “просто ще одна змінна версія” - це новий адаптер візуалізації для ioBroker, який базується на фреймворці Material UI для веб-розробки. Візуалізація має власну структуру пристроїв та модулів, що означає, що у візуалізацію можна гнучко інтегрувати широкий спектр пристроїв та загальних точок даних. Візуалізація налаштовується за допомогою вкладок та стовпців. Потім пристрої можна легко вставити у візуалізацію за допомогою так званих віджетів.

Таблиця 3.1 – Порівняння середовищ візуалізації ioBroker.

Тип візуалізації	Складність	Власний дизайн	Власні віджети	Комплексність	Адаптивний дизайн	Спільнота
VIS	Високий	Так	Так	Високий	Ні	Дуже добре
Material UI	Середній	Ні	Ні	Середній	Так	Добре
iQuontrol	Середній	Ні	Ні	Середній	Так	-
HABPanel	Середній	Ні	Так	Середній	Ні	-
Lovelace-UI	Середній	Так	Так	Середній	Так	-
TileBoard	Середній	Так	Так	Середній	Так	-
Node-RED	Високий	Так	Так	Високий	Так	Добре
Jarvis	Середній	Ні	Ні	Середній	Так	Добре

Зараз я перерахував усі візуалізації для ioBroker і показав, як працює кожна візуалізація. Я друг візуалізації VIS, оскільки я справді можу використати всю свободу дизайну та всі технічні можливості. Однак для мене найбільшим недоліком є дуже жорстке створення візуалізації для кінцевого пристрою (дозвіл дисплея). Тут вигляд не масштабований, а тому не пристосовується до інших дисплеїв. З цієї причини я почав відтворювати свою існуючу візуалізацію на основі VIS за допомогою динамічного адаптера. Я зробив свої перші кроки та досвід із HabPanel. Через відсутність віджетів та проблеми з масштабуванням до різної роздільної здатності, я протестував Material-UI та Lovelace на наступному кроці. Ці два адаптери підтримують адаптивний дизайн, і тому створена візуалізація може чудово відображатися на різних пристроях. Material-UI та Lovelace відрізняються між собою розташуванням та з'єднанням точок даних. У Lovelace точки даних в об'єктах повинні бути звільнені спеціально для Lovelace. Звичайно, це додаткові зусилля, але я вважаю візуалізацію справді вдалою та зрозумілою. З цієї причини я б зараз рекомендував VIS для складної візуалізації.

3.2 Архітектура та структура проекту

Adapter Admin використовується для управління всією установкою ioBroker. Він надає веб-інтерфейс. Це називається <IP-Adresse des Servers>: 8081.

Цей адаптер створюється безпосередньо під час установки ioBroker, установка вручну не потрібно

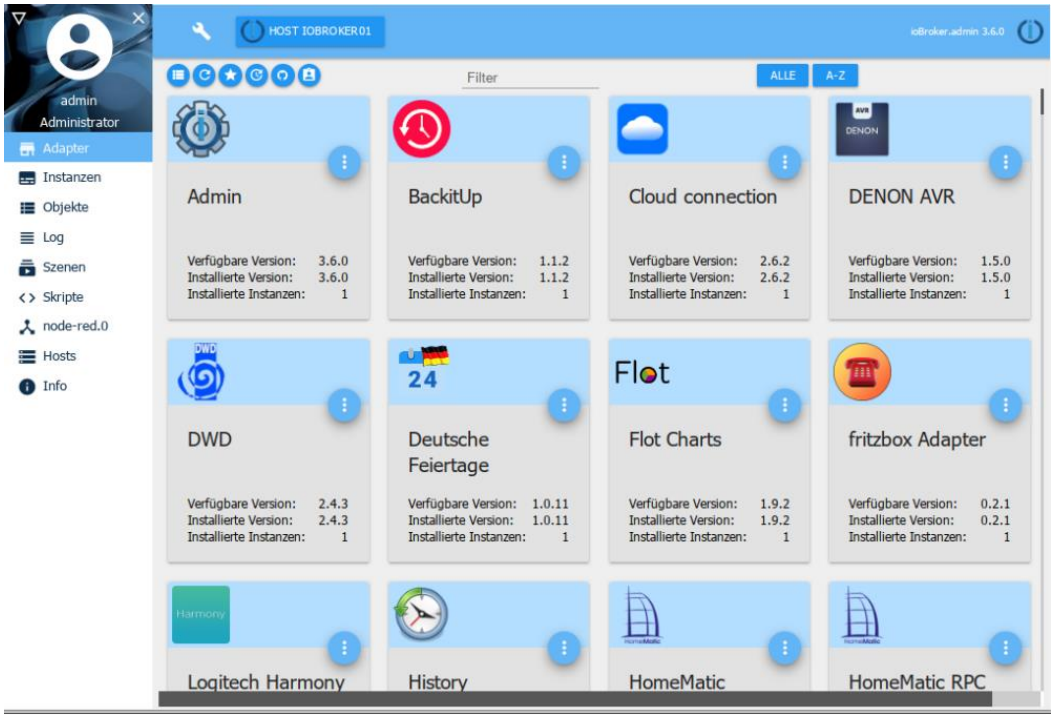


Рисунок 3.7 – Архітектура IoBroker

GUI, що надається адаптером, може включати, але не обмежується: отримані наступні функції:

- Введіть загальносистемні налаштування
- Установка додаткових адаптерів і примірників
- Доступ до конфігурації примірників
- Доступ до огляду об'єкта
- Доступ до огляду стану об'єктів
- Доступ до адміністрування користувачів і груп
- Доступ до лог-файлу
- Адміністрація господарів

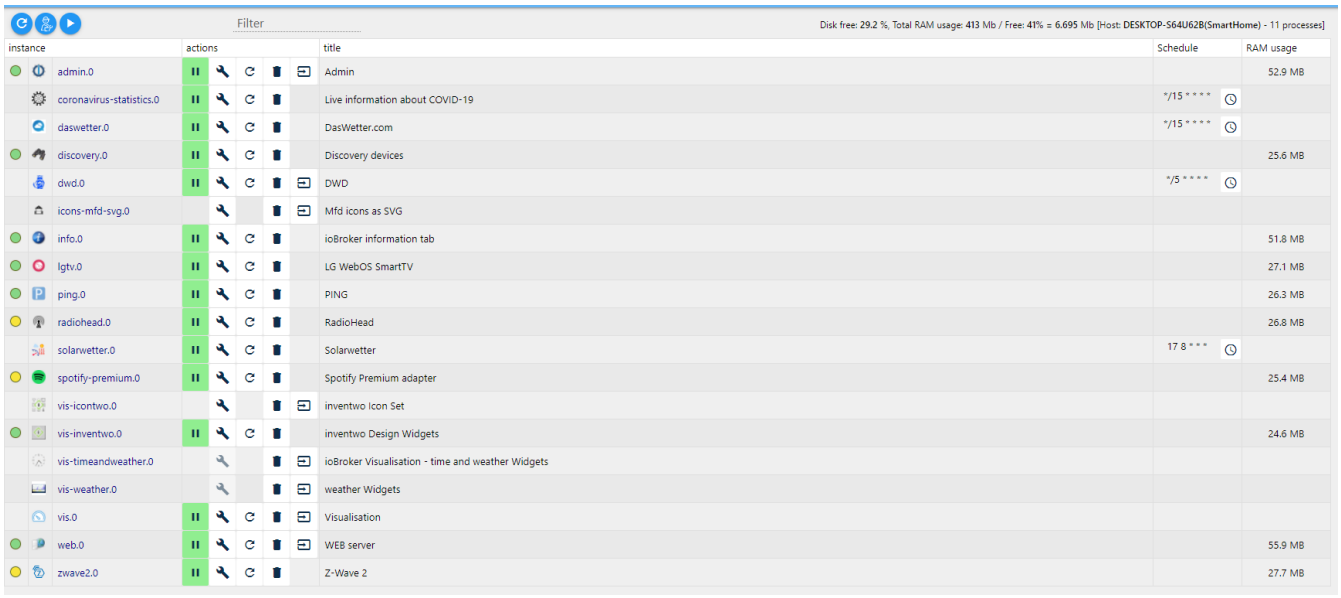


Рисунок 3.8 – Перелік частини адаптерів використаних в IoBroker

Тепер розглянемо використанні адаптери для створення власної візуалізації процесів та пристроїв. Розглянемо адаптер VIS Material Design.

За допомогою віджетів Material Design Widgets адаптера VIS ви можете створити реальну візуалізацію дизайну матеріалів за допомогою VIS. У серії статей ми покроково встановимо адаптер і створимо першу візуалізацію дизайну матеріалу.

Пізніше навігаційне меню повинно відображати таку структуру:

– Огляд
– освітлення
– статус
– система
– ioBroker
– гомематичний
– відтінок

Тепер ми відкриваємо проект VIS, а на першому кроці створюємо інші уявлення. На наступному кроці відкриваємо індекс перегляду, у якому зберігається віджет навігації Top Bar Bar. Тепер ми позначаємо віджет і розширюємо кількість елементів навігації до 3. Потім підменю розміщується в третьому записі. Для визначення підменю ми також змінюємо кількість підменю на 3 (три записи підменю).

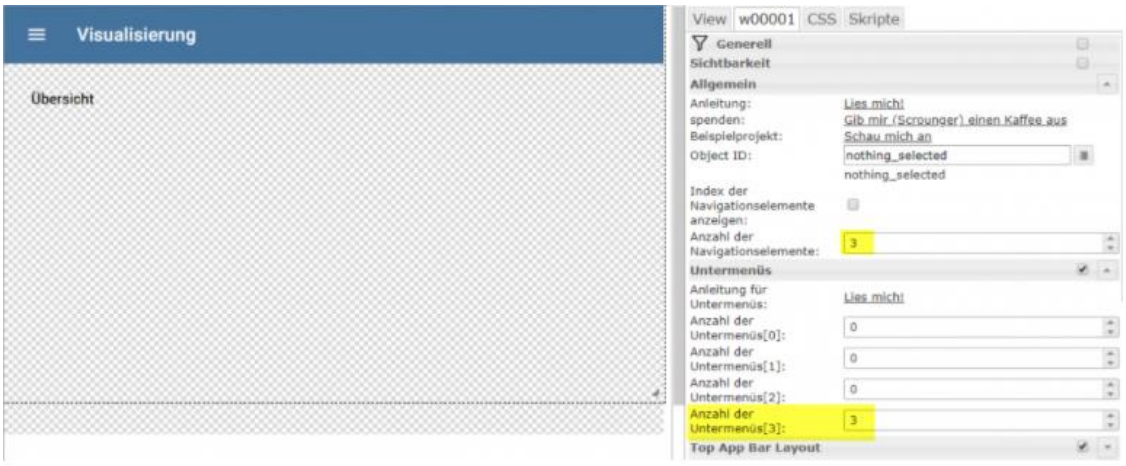


Рисунок 3.9 – Перегляд Material Design Widgets

Позначення підменю контролюється за допомогою об'єкта JSON. Основний елемент позначений властивістю itemText. Список підпунктів містить назви окремих елементів підменю. Зверніть увагу, що окремі записи починаються з квадратних дужок і розділяються комами (приклад ["A", "B", "C"]).

Зараз ми завершили конфігурацію віджета Top Top Bar Navigation і тепер можемо розширити віджет View in View на View index на наступному кроці. У властивостях віджета ми тепер міняємо кількість переглядів (кількість значень до) на 6, що представляє в цілому 7 переглядів (0-6). Тепер у нових полях для подань ми зберігаємо подання, створені на початку. Я також використовував ioBroker View для основного входу в систему тут.

3.3 Опис та розширений розбір роботи

Було розробленно додаток VIS, який включає в себе управління пристроями, а також синхронізованно з системою розумного будинку Z-Wave. Саме розглянемо візуалізацію управління термостатами. Особливості цієї візуалізації:

- Потрібна лише одна сторона для будь-якої кількості кімнат та профілів
- Логічний поділ на різні "картки", щоб забезпечити адаптивний дизайн.
- Більше віджетів, що перекриваються.
- Кольори (шрифт, фон тощо) можна встановити за допомогою діалогового вікна.
- Незалежна адаптація Vis при зміні конфігурації адаптера (кількість профілів, DecreaseMode, тип профілю, періоди).
- Додаткова картка з оглядом стану вікон кожної кімнати.

– Чуйний дизайн.
– Складані елементи для економії місця.
– Простіша установка навіть для початківців, оскільки це цілісний, незалежний проект Vis, який також підходить як основа / приклад для вашого власного загального проекту.
– Додаткова зразок сторінки статусу кімнати для власного розширення основних функцій.

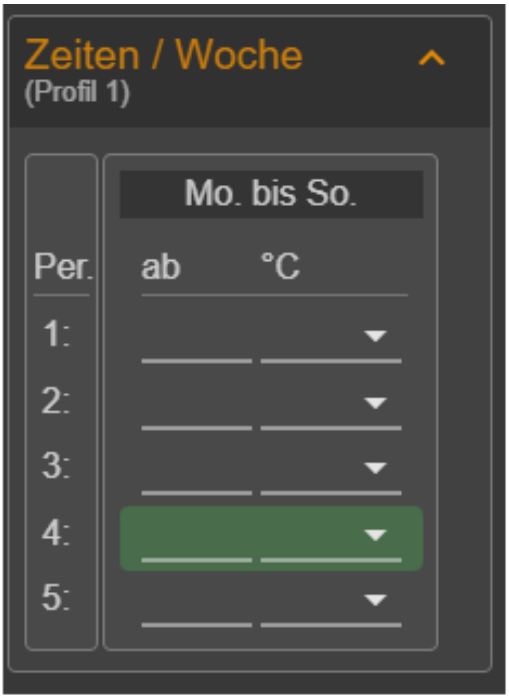


Рисунок 3.10 – Для профілю "Усі дні разом"

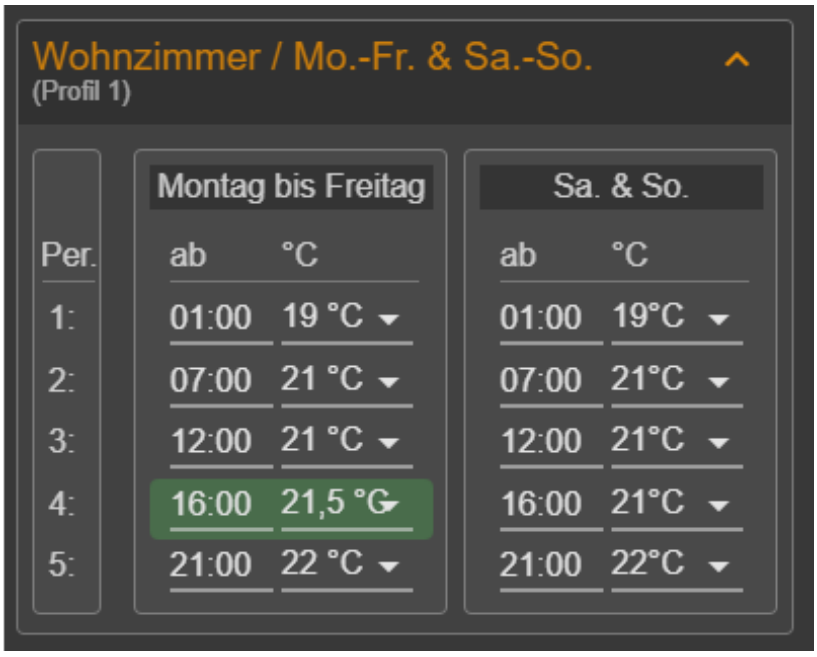


Рисунок 3.11 – З типом профілю "Пн-Пт / Сб-Нд"

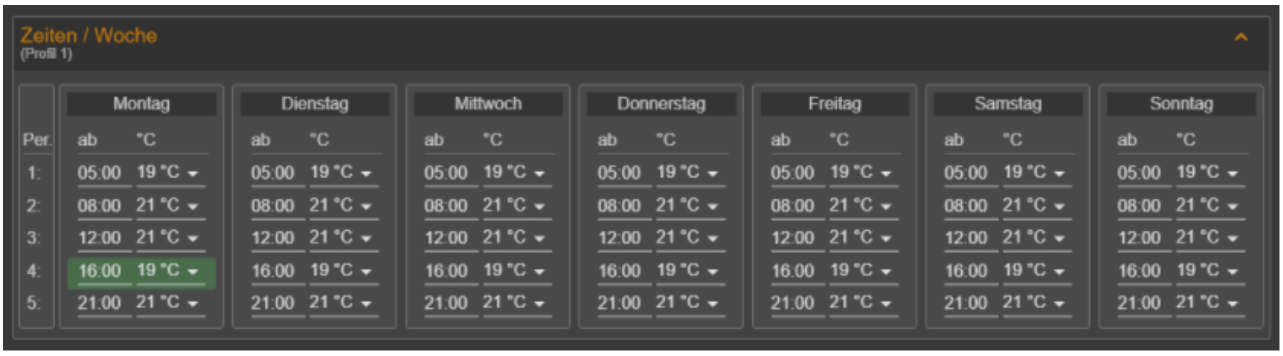


Рисунок 3.12 – 3 типом профілю "кожен день відокремлено"

Шаблон простір для карт

Служить шаблоном макета, якщо потрібно відобразити основні значення для кожної кімнати. Ви повинні бути повністю налаштовані вами. Найкращий спосіб діяти наступним чином: Позначте всі віджети клаві7шею Ctrl-A, натисніть Експорт віджетів та скопіюйте відображений код у буфер обміну (Ctrl-A / Ctrl-C). Тепер створюється новий вигляд і називаємо його вітальня. Тепер у вас є порожня сторінка, тут ви вибираєте віджети імпорту та вставляєте код із буфера обміну за допомогою Ctrl-V. Тепер ця сторінка має всі елементи сторінки шаблону. Ви повторюєте цей процес для кожної кімнати. Після того, як ви створили всі сторінки, ви знову переглядаєте їх по черзі та коригуєте ідентифікатори об’єктів віджетів.

Тепер, коли ви створили всі картки, вам доведеться інтегрувати їх на сторінку. Це робиться через сторінку "contHzng". Це центральна сторінка, яка визначає, яка картка відображається в якому розмірі та порядку; Для цього використовуються віджети "Перегляд у віджеті". Тепер ви шукаєте віджет, який показує картку шаблону кімнати, позначаєте його та копіюєте (Ctrl-C). Тепер ви вставляєте цю копію за допомогою Ctrl-V і міняєте два місця; За допомогою "Viewname" ви вибираєте перший із власних виглядів кімнати. У "класі CSS" ви змінюєте частину "mdui-order-30" на "mdui-order-35". На наступному перегляді потім 40 і так далі. Ви використовуєте його для визначення порядку відображення. Етапи 5 спрощують додавання чи перестановку чогось, тут ви можете використовувати будь-який бажаний розмір кроку.

У кінцевому випадку, результат вийшов схожим на те що зображено на малюнку 3.12.

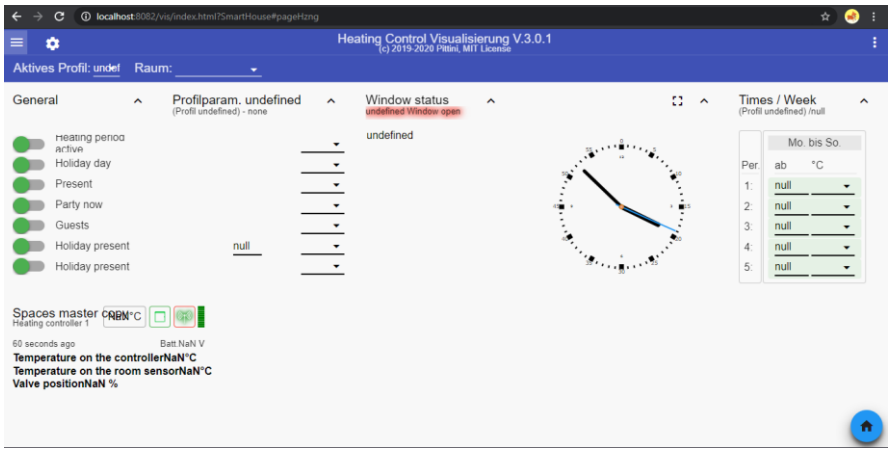


Рисунок 3.13 – Вигляд додатку управління термостатами

Висновоки до розділу

В цьому розділі проробленно візуалізацію управління розумним будинком за домогою VIS додатку, який підтримується на платформі ioBroker. Використано існуючі адаптери, а саме: bars Widgets, canvas-gauges style Widgets, MFD icons as PNG/SVG, fancyswitch style, time and weather Widgets, Admin, BackItUp, Discovery devices, Manage devices, Monitoring and Maintenance, Web server, DasWetter.com тощо.

Найвливовішими та найвикористовуваніші є адаптери: Material Design Widgets та LG WebOs SmartTV. Усі потрібні та росширенні данні та інформація адаптерів знаходиться у додтаку Г.

РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЄКТУ

4.1. Опис ідеї проєкту

Таблиця 4.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Система моніторингу та оцінки загроз інформаційній безпеці розумного будинку	Використання додатку у системах розумного будинку для забезпечення інформаційної безпеки	Функціональна потужність та можливість налаштування системи від індивідуальні потреби, простота експлуатації, яка не потребує спеціальних знань та навичок, відносно невисока ціна продукту

Таблиця 4.2. Опис ідеї стартап-проекту

№	Техніко-економічні характери-тики ідеї	Продукція конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проєкт	Dojo	CUJO	Bitdefender Box			
1	Якість дизайну	С	В	В	С		+	
2	Зручність використання	С	В	В	В	+		
3	Вимоги до системи	Н	С	С	С			+
4	Можливість настройки роботи	В	Н	Н	Н			+
5	Функціональна потужність	В	Н	Н	С			+

4.2. Технологічний аудит ідеї проекту

Таблиця 4.3. Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Розробка системи Система моніторингу та оцінки загроз інформаційній безпеці розумного будинку	Використання інструмента Windows Presentation Foundation (WPF)	Наявна. Має велику кількість готових програмних інструментів для вирішення різних проблем. Має гарно розроблені бібліотеки, конструктор.	Вільна
		Використання мови програмування C#	Наявна. Має основні простори імен для модуля	Вільна
		Використання мови програмування Python	Наявна. Має один фреймворк	Вільна
Обрана технологія реалізації ідеї проекту: WPF, мова програмування: C#				

Система WPF обрана через можливості використання будь-якої .NET-сумісної мови програмування разом з мовою XAML. WPF і XAML об'єднуються в повнофункціональну систему подання для створення візуально привабливих класичних додатків Windows, що включають в себе користувальницький інтерфейс, мультимедіа та складні бізнес-моделі.

Мова програмування C # має кілька класів для ефективної і швидкої роботи з XML-документами, які обираються в залежності від поставлених завдань.

4.3. Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 4.4. Попередня характеристика потенційного ринку

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	4
2	Загальний обсяг продаж, грн./ум.од	100 тис. долларів США на рік
3	Динаміка ринку	Ринок систем безпеки для розумного будинку зростає з кожним роком
4	Наявність обмежень для входу	Відсутні, відкритий ринок
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі або по ринку, %	70%

Висновок: враховуючи кількість головних гравців по ринку, зростаючу динаміку ринку, невелику кількість конкурентів та середню норму рентабельності можна зробити висновок, що на даний момент, ринок для входження стартап-продукту є привабливим.

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проєкту

№№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	Вимоги споживачів до товару
11	Потреба захисту інформаційної безпеки у розумному будинку	Фізичні особи	Як правило, користуються системами розумного будинку з низьким рівнем захисту інформації	Зручність у використанні, точність роботи, надійність, швидка робота системи. Спроможність швидко освоїтись як користуватись системою.
		Підприємства	Використовують тільки системи, вироблені спеціалізованими компаніями, з достатнім рівнем захисту;	

Таблиця 4.6. Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренти	Наявність конкурентів котрі надають схожі рішення	Зменшення ціни на поставлену послугу; Розробка унікальних характеристик товару; Надання ліцензій на обслуговування
2	Кошти на розробку та підтримку продукту	Закінчення грошей та недостатнє фінансування	Залучення додаткових інвесторів, мотивація роботи на перспективу; Ітеративна розробка продукту задля покрокового виведення продукту на ринок та отримання відповіді користувачів
3	Вихід аналогу	Вихід аналогу даного товару може призвести до знецінення та безідейності даного товару	Вихід товару на ринок в коротші строки з не повною, але достатньою, функціональністю для зацікавлення усіх цільових аудиторій; Проведення рекламної компанії

Таблиця 4.7. Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Новий продукт	Вихід на ринок, Зменшення монополії, Надання нових рішень у сфері	Розробка нової функціональності; Вихід нової продукції на ринок; Надання різноманітних типів ліцензій в залежності від потреб користувача \ замовника.
2	Вихід аналогу	Надати продукт з певними характеристиками та можливостями що відсутні у компаній конкурентів	Аналіз ринку та користувачів задля задоволення їх потреб та надання функціональності у найкоротші строки за ціну, котра є дешевшою ніж у продуктів-замінників.

Закінчення таблиці 4.7.

3	Зворотній зв’язок від користувачів	Можливість отримання необхідної інформації для вдосконалення продукту	Наявність вхідних даних та реакція на них з боку команди розробників задля задоволення потреб та бажань кінцевих користувачів системи моніторингу та оцінки загроз розумного будинку.
4	Грошова винагорода за рекламу	При достатньому попиту на систему моніторингу та оцінки загроз інформаційній безпеці можлива комерціалізація продукту на основі реклами задля отримання грошової винагороди для подальшого розвиту продукту та оплати заробітної плати працівникам	Точкова комерціалізація продукту; Введення реклами; Ведення додаткових коштів у проект задля його подальшого розвитку.

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Тип конкуренції: монополістична	Товар від кожної компанії на ринку, являється недосконалим замінником товару, реалізованого іншими фірмами; На ринку є умови для входу та виходу; Ціна корелює між суперниками;	Розробка продукту з характеристиками, які покривають сфери вживання що не покривають інші товари-замінники; Кореляція цін у відповідності до товарів замінників; Різни типи ліцензій.

Закінчення таблиці 4.8.

2	Рівень конкурентної боротьби: світовий	Всі продукти замінники розроблялись інтернаціональними командами з різних куточків світу, продукти не належать до певної держави, а належать команді розробників	Вихід на ринок збуту продукту з клієнто-необхідною функціональністю; Налагодження маркетингу на основних Інтернет ресурсах задля охоплення великої кількості потенційних користувачів; Надання бета-версій продукту.
3	Галузева ознака: внутрішньогалузева	Даний тип продукту може використовуватися тільки у сфері розробки IT додатків \ продуктів	Надання зручного, інтуїтивно зрозумілого інтерфейсу; Підтримка всім відомих методів взаємодії з середовищем розробки; Наявність документації та онлайн підтримки.
4	Конкуренція за видами товарів: товарно-видова	Дана конкуренція – конкуренція між товарами одного виду.	Впровадження функціональності яка відсутня у товарів-замінників; Спрощення інтерфейсів; Надання підтримки.
5	Характер конкурентних переваг: цінова та не цінова	Цінові переваги – точкова комерціалізація; Не цінова – надання функціональності, що відсутня у товарах-замінниках.	Надання платних ліцензій лише на критично важливу функціональність для клієнта з певним строком підтримки, що зазначена у відповідній ліцензії; Впровадження унікальної функціональності.
6	За інтенсивністю: марочна	Наявність унікального знаку що відрізняє даний продукт від продуктів-замінників	Впровадження власної назви та власного знаку.

Таблиця 4.10. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Середня	Низька	Низька	Середня	Середня
Висновки	На даний час конкурентів в даній області невелика кількість в порівнянні з розмірами ринку клієнтів для системи	Можлива поява нових подібних розробок для, але прогрес у цьому напрямку не дуже стрімкий	Немає залежності від постачальників, так як використовується програмне забезпечення, яке розроблене власноруч без додаткових постачань від третіх осіб	Клієнти можуть диктувати умови на ринку через повідомлення на форумах або в полі відгуків в точках продажу додатку	Середня ймовірність появи нових систем, можливість введення стандартизації систем інформаційної безпеки в розумному будинку

Проаналізувавши можливості роботи на ринку з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з існуючих продуктів не впливає у великій мірі на поточну ситуацію на ринку в цілому, кожний з існуючих продуктів має свою специфічну сферу використання та свої позитивні та негативні сторони щодо рішення певних типів задач, то робота та вихід на даний ринок є можливою і реалізованою задачею.

Для виходу на ринок продукт повинен мати функціонал що відсутній у продуктів-аналогів, повинен задовольняти потреби користувачів, мати необхідний та достатній функціонал з конфігурування, підтримку зі сторони розробників та можливість розробки спеціального функціоналу за відповідною ліцензією.

Таблиця 4.11. Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Прагматичність	Через запуск стартапу система буде не дуже складно з точки зору архітектури перший час. Через певний період із додаванням функціоналу та оптимізації алгоритмів роботи програмний код буде все складнішим. Такий етап наступить не раніше одного року постійної роботи над проектом.
2	Зручність	Оскільки стартап розробляється на багатьох платформах з різною шириною екранів, то зручність використання системи на різних пристроях буде відігравати не малу роль у спроможності конкурувати з іншими гравцями ринку
3	Швидкість роботи	Швидкість роботи відіграє велику роль для користувачів, оскільки вони не будуть готові чекати декілька хвилин на виведення результату роботи додатку.
4	Оптимізація	Якщо додаток буде дуже часто видавати помилки при роботі, то користувачі не будуть вважати додаток надійним
5	Приватність	В останні роки приватність людей та інформація щодо них все частіше зловживається шахраями або великими корпораціями, які потребують погодження з умовами доступу до приватної інформації та її обробки.
6	Технічна підтримка	Якщо технічна підтримка компанії буде працювати своєчасно та швидко, то це допоможе зберегти репутацію компанії на відміну від конкурентів, де їй не приділяють увагу.

Таблиця 4.12. Порівняльний аналіз сильних та слабких сторін системи

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з запропонованим						
			-3	-2	-1	0	+1	+2	+3
11	Зручність використання	17	B+			+			
12	Якість дизайну	12					++		
33	Можливість налаштування системи	18	+		++				
44	Функціональна потужність	17		+		+	+		
55	Простота експлуатації	13					+		+
66	Якість та зрозумілість інтерфейсу	13			+		+		

SWOT-аналіз дає чітке уявлення про фактори зовнішнього і внутрішнього середовища і вказує, в яких напрямках потрібно діяти, використовуючи сильні сторони, щоб максимізувати можливості і звести до мінімуму загрози і слабкі сторони. За допомогою цього методу можна позначити основні проблеми проєкту, визначити шляхи вирішення і перспективу розвитку.

Таблиця 4.13. SWOT аналіз стартап-проекту

Сильні сторони (S): – Функціональна потужність; – Можливість налаштування під індивідуальні вимоги споживача; – Відносно невисока ціна. – Відсутність додаткового збору даних.	Слабкі сторони (W): – Невисока якість дизайну; – Недостатня простота експлуатації.
Можливості (O): – Зростання кількості користувачів систем розумного будинку для житлових будинків. – Підвищення обізнаності користувачів систем розумного будинку про необхідність захисту інформаційної безпеки їх систем.	Загрози (T): – Відсутність попиту на дану розробку. – Жорсткість вимог в законодавстві щодо програмних продуктів та інформаційних технологій. – Розвиток конкурентних розробок.

Таким чином, в результаті SWOT-аналізу були розглянуті сильні і слабкі сторони розробки, можливості та можливі загрози. Основні слабкі сторони наукової розробки це низька якість дизайну і складність експлуатації. У першому випадку при будь-яких можливостях та загрозах слід звернутися до дизайнера. У другому - постаратися спростити експлуатацію продукту. Крім зміни попиту одночасно джерелом можливостей і загроз є законодавство: введення обмеження для іноземного ПЗ може значно підвищити попит на розробку, знизивши конкуренцію, посилення вимог для розроблюваних ПЗ може вимагати переробки або доопрацювання частини функціоналу продукту.

Таблиця 4.14. Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Безкоштовне надання певного функціоналу у користування споживачам на обмежений термін	Головний ресурс – люди, даний ресурс - наявний	2-3 місяці
2	Реклама	Залучення власних коштів для реклами товару	1-2 місяці
3	Написання статей та опис товару на відомих ресурсах	Головний ресурс – час, даний ресурс - наявний	2-3 тижні
4	Презентація товару на хакатонах й інших ІТ заходах	Ресурс – час та гроші для участі, наявні	1-3 місяці

4.4. Розроблення ринкової стратегії проекту

Таблиця 4.15. Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Домовласники /приватні особи	Присутня	Середній	Присутня	Легка
2	Підприємства	Присутня	Середній	Присутня	Складна
Які цільові групи обрано: 1					

Відповідно до проведеного аналізу можна зробити висновок, що розроблюваний продукт призначений для особистого використання фізичними особами в системах розумного будинку будь-якого виду. Відповідно до стратегії охоплення ринку збуту товару обрано стратегії цільового маркетингу та особистого контакту.

Таблиця 4.16. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Надання функціональності що відсутня у товарів-замінників, підтримка клієнтів	Проведення реклами, освітлення унікальної функціональності через інтернет ресурси та інші канали, контакт напряму з споживачами;	Зниження ступеню замінності товару; Прихильність клієнтів; Відмітні властивості товару; Відмітні характеристики товару;	Стратегія диференціації

Таблиця 4.17. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
Ні, оскільки є товари-замінники, але дані товари замінники не мають деякого необхідного функціоналу	Так, ціль компанії знайти нових споживачів та, частково, забрати існуючих у конкурентів задля задоволення потреб останніх	Компанія частково копіює характеристики товару конкурента, основна ціль компанії розробка нового унікального функціоналу, з підтримкою основного функціоналу конкурентів	Стратегія заняття конкурентної ніші

Таблиця 4.18. Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Ефективність	Диференціація	Система може бути використана для забезпечення інформаційної безпеки в будь-яких системах розумного будинку	Висока ефективність в оцінюванні загроз інформаційній безпеці
2	Відкритість вихідного коду	Диференціація	Перспектива розвитку проекту	Розвиток в науці
3	Приватність	Заняття конкурентної ніші	Ваші дані належать тільки вам	Захищеність особистої інформації

Відповідно до проведеного аналізу можна зробити висновок, що стартап-компанія вибирає як базову стратегію розвитку – стратегію диференціації, як базову стратегію конкурентної поведінки – стратегію заняття конкурентної ніші.

4.5 Розроблення маркетингової програми стартап-проекту

Таблиця 4.19. Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Функціональність	Можливість налаштування додатку під індивідуальні потреби користувача	Можливість використовувати з будь-якою системою розумного будинку
2	Зручність у використанні	Зрозумілий інтерфейс	Можливість автоматично визначити склад системи та її обмеження

Таблиця 4.20. Опис трьох рівнів моделі товару

<table>
<tr><td>Рівні товару</td><td colspan="3">Сутність та складові</td></tr>
<tr><td>1. Товар за задумом</td><td colspan="3">Система моніторингу та оцінки загроз інформаційній безпеці розумного будинку</td></tr>
<tr><td rowspan="8">2. Товар у реальному виконанні</td><td>Властивості/характеристики</td><td>М/Нм</td><td>Вр/Тх/Тл/Е/Ор</td></tr>
<tr><td>Зручність</td><td>Нм</td><td>Е</td></tr>
<tr><td>Швидкість роботи</td><td>Нм</td><td>Тх</td></tr>
<tr><td>Оптимізація</td><td>Нм</td><td>Тх</td></tr>
<tr><td>Технічна підтримка</td><td>Нм</td><td>Тх</td></tr>
<tr><td>Приватність</td><td>Нм</td><td>Тх</td></tr>
<tr><td colspan="3">Якість: дотримання загальновживаних стандартів, нормативів</td></tr>
<tr><td colspan="3">Пакування: ліцензія на використання системи</td></tr>
<tr><td rowspan="2">3. Товар із підкріпленням</td><td colspan="3">До продажу: наявна повна документація, акції на придбання декількох ліцензій, знижки для певних сегментів на покупку товару</td></tr>
<tr><td colspan="3">Після продажу: додаткова підтримка спеціалістів налаштування, підтримка з боку розробника</td></tr>
<tr><td colspan="4">За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності, патент</td></tr>
</table>

Таблиця 4.21. Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
200-600$	300-1200$	600-10000$/міс	200-400$

Таблиця 4.22. Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Всі користувачі будуть купувати товар поодинці	Можливість скачувати додаток в будь-який час, в будь-якому місці	2 рівня (посередник+клієнт)	Роздріб

Таблиця 4.23. Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Домовласники	Інтернет, конференції, приватні зустрічі	Безпека	Показати можливість користування	Рекламне звернення спрямовано до потенційних клієнтів, користування системою

Як результат було створено ринкову (маркетингову) програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

Висновки по розділу

В четвертому розділі описано стратегії та підходи з розроблення стартап-проекту, а саме: описано ідею проекту, розглянуто технологічний аудит ідеї проекту, технологічну здійсненність ідеї проекту, проведено аналіз ринкових можливостей –

дано попередню характеристику потенційного ринку та характеристику потенційних клієнтів проекту.

Описано фактори загроз, фактори можливостей, зроблено ступеневий аналіз конкуренції на ринку, аналіз конкуренції в галузі за М. Портером. Обгрунтовано фактори конкурентоспроможності, зроблено порівняльний аналіз сильних та слабких сторін додатку.

Проведено SWOT-аналіз проекту, визначено альтернативи ринкового впровадження. Розроблено ринкову стратегію проекту: обрано цільові групи потенційних споживачів, визначено базову стратегію розвитку, базову стратегію конкурентної поведінки, визначено стратегію позиціонування.

Розроблено маркетингову програму стартап-проекту: визначено ключові переваги концепції потенційного товару, описано три рівні моделі товару, визначено межі встановлення ціни та зформовано систему збуту. Описано концепцію маркетингових комунікацій..

ВИСНОВКИ

В процесі виконання роботи вирішена проблема зручності використання додатків, котрі надаються розробниками існуючих систем розумного будинку. Набагато зручніше доповнювати систему розумного будинку Z Wave з іншими пристроями, котрі не підтримуються протоколами існуючих протоколів розумних будинків.

Так як, вже існує система управління розумним будинком, котра вискористовує протокол Z-Wave та має у системі вже: головний котроллер під назвою Z-Wave Vera Plus, 4 термостати управлінням температурою електропідлоги. У другому розділі описанно стійкість протоколу Z-Wave від сторонніх чиників та спроб злому системи (DDoS, hack та інші). Переконавшись в тому що протокол Z-Wave надійний у користуванні та долучанні пристроїв розумного будинку, вирішенно знайти систему автоматизації, яка підтримує роботу протоколу Z-Wave.

Порівнянно 5 способів створення візуалізації та взаємодії між пристроями, а саме: створення способу рішення за допомогою «справжнього» розумного будинку, система автоматизації zVirtualScences, система автоматизації, система автоматизації Ago Conrol, система автоматизації ioBroker, система автоматизації Domoticz. Дві системи автоматизації ioBroker та Domoticz по іх існуючим функціям та відгукам користувачів та розробників систем є найкращими у своєму роді. Domoticz прийнято не використовувати, тому що, у цій системі автоматизації існує недоліки, один із головних, це те що не можна використовувати цю систему безкоштовно у своїх не комерційних цілях. Посперичавшись було прийнято використовувати платформу ioBroker. Для розробників систем та візуаліцій існує пільги, а також форум, де можна спілкуватися та обмінюватися інформацією з людьми.

Об’єм робіт зроблений за допомогою платформи ioBroker, використавши адаптери OpenZWave, zwave-js та створивши візуалізацію за допомогою адаптера «VIS візуалізація».

Налаштувавши всі адаптери, було створенно 4 головні сторінки додатку, котрі завантаженні на сервер Google, та виконуються у реальному часі. Сторінки мають вмістку: головне меню, статус по коронавірусу та термостати. Також створенна підтримка користування інтерфейсом з різних пристроїв (телефон, комп’ютер,

планшет тощо). Усі налаштування та вигляд створенного інтерфейсу можна побачити на додатках.

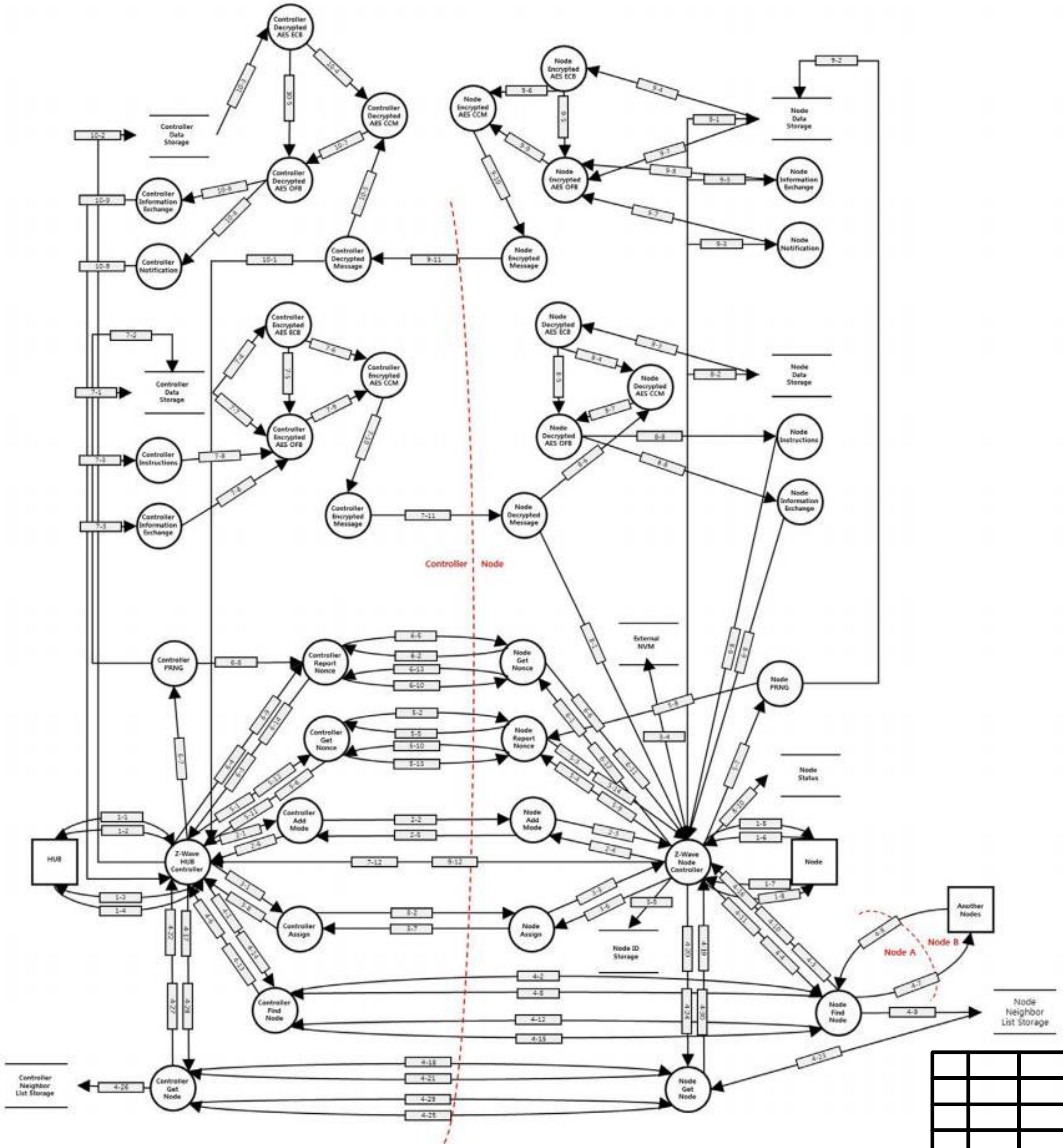
ПЕРЕЛІК ПОСИЛАНЬ

1. S. Giordano, D. Puccinelli, When sensing goes pervasive, Pervasive Mob. Comput. 17 (2015) 175–183.
2. C. Badenhop, J. Fuller, J. Hall, B. Ramsey, M. Rice, Evaluating IT-T g.9959 Ubased wireless systems used in critical infrastructure assets, Crit. Infrastruct. Prot. IX (2015) 209–227
3. http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol413/paper12.pdf
4. K. Hoskins, Security Vulnerabilities in Z-Wave Home Automation Protocol, Tufts University Department of Computer Science, 2016, pp. 1–14
5. https://github.com/OpenZWave/open-zwave
6. https://www.domoticz.com/DomoticzManual.pdf
7. http://www.opensourceautomation.com/wiki/index.php?title=Main_Page
8. http://aarondrabeck.github.io/zVirtualScenes/scripting.html
9. https://www.iobroker.net/?page_id=66&lang=en#ru/documentation/admin/README.md
10. https://www.smarthome-tricks.de/software-iobroker/
11. https://z-wave.me/essentials/ZWayManual.pdf

ДОДАТКИ

ДОДАТОК А.

Блок-схеми протоколу Z-Wave рівень 2



					ІК-91.18 3132.02			
Зм.	Лист	№ докум	Підпис	Дата	Блок-схеми протоколу Z-Wave рівень 2	Літ.	Маса	Мірило
Розроб.		Лємешко А.Д.						
Перев.		Лісовиченко О.І.				Лист	Листів	
Н. контроль		Пасько В.П.			Кафедра Технічної кібернетики	Група ІК-91мп		
Затв.		Пархомей І.Р.						

Підпис і дата

Інв. № дубл.

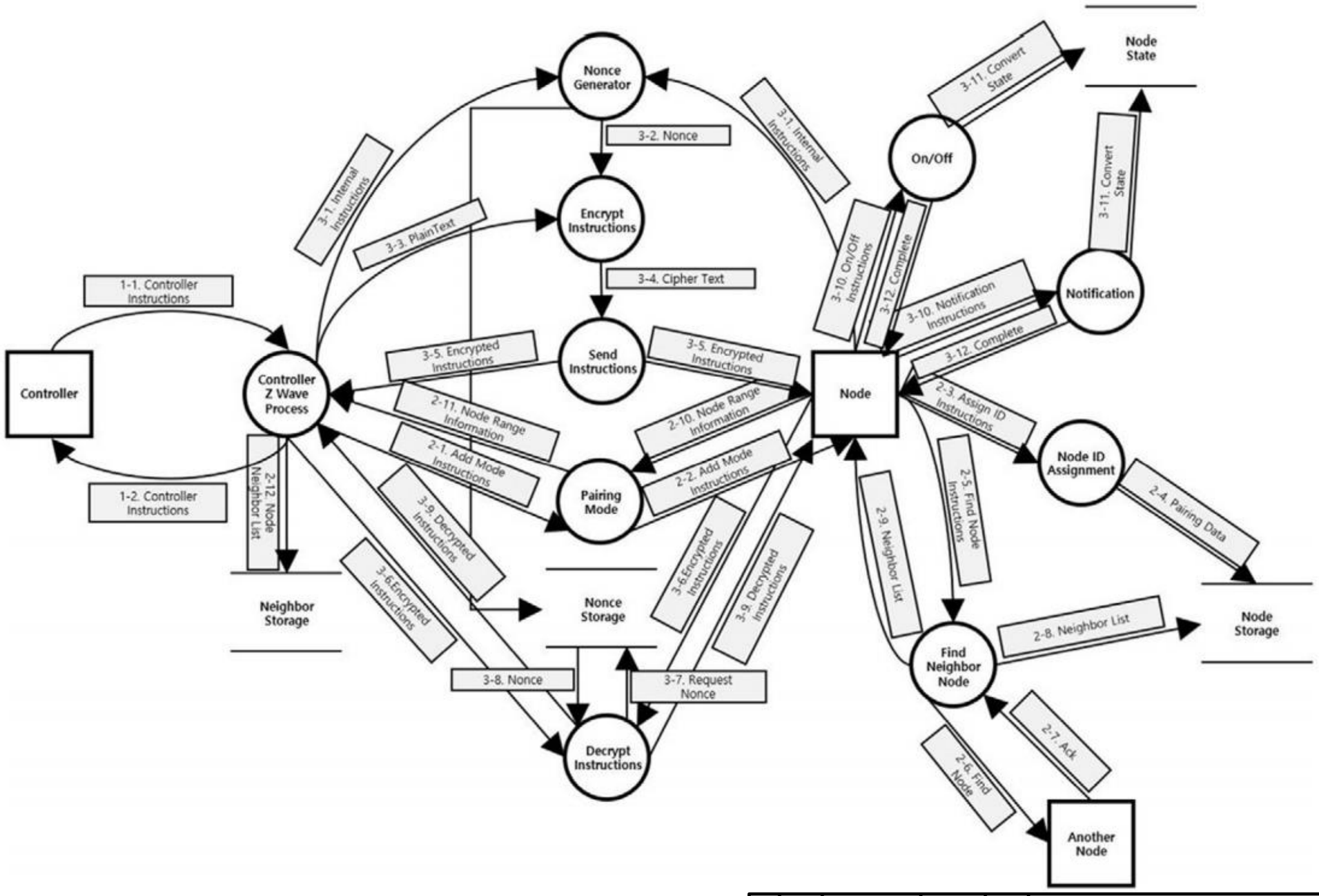
Взам. інв. №

Підпис і дата

Інв. № ориг.

ДОДАТОК Б

Діаграма протоколу Z-Wave рівень 1



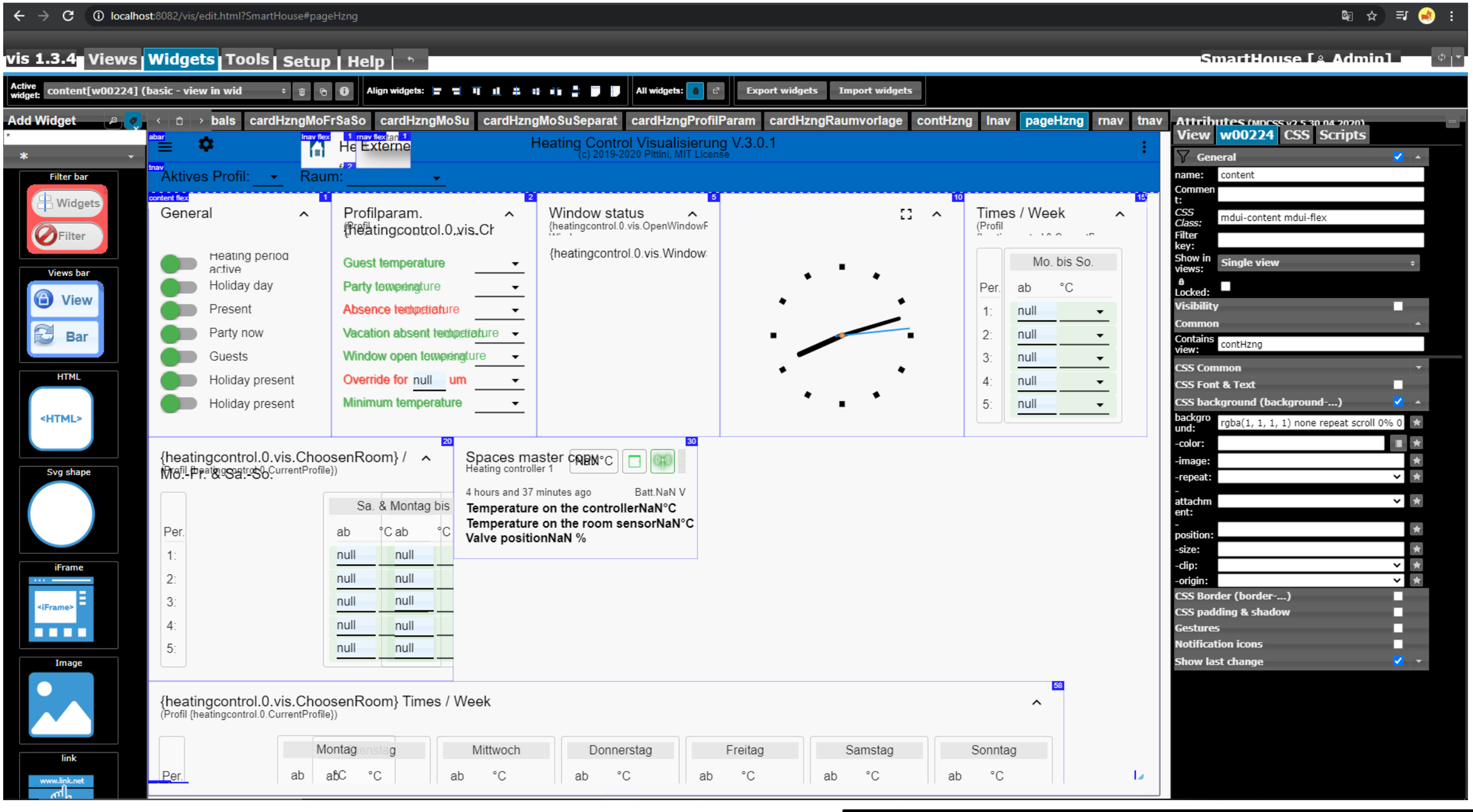
					ІК-91.18 3132.03	
Зм.	Лист	№ докум	Підпис	Дата	Діаграма протоколу Z-Wave рівень 1	Літ. / Маса / Мірило
Розроб.		Лємешко А.Д.				
Перев.		Лісовиченко О.І.				Лист / Листів
Н. контроль		Пасько В.П.			Кафедра Технічної кібернетики	Група ІК-91мп
Затв.		Пархомей І.Р.				

Підпис і дата | Інв. № дубл. | Взам. інв. № | Підпис і дата | Інв. № ориг.

ДОДАТОК В

Віджети управлінням термостатами (ioBroker-vis)

Віджети управлінням термостатами (ioBroker-vis)



Демонстраційний плакат №1
до дипломної роботи на тему
„Система управління розумним будинком”

Розробив: Лємешко А.Д.
Прийняв: Лісовиченко О.І.

ДОДАТОК Г

Інтерфейс управління телевізором (lgtv.0)

Інтерфейс управління телевізором (lgtv.0)

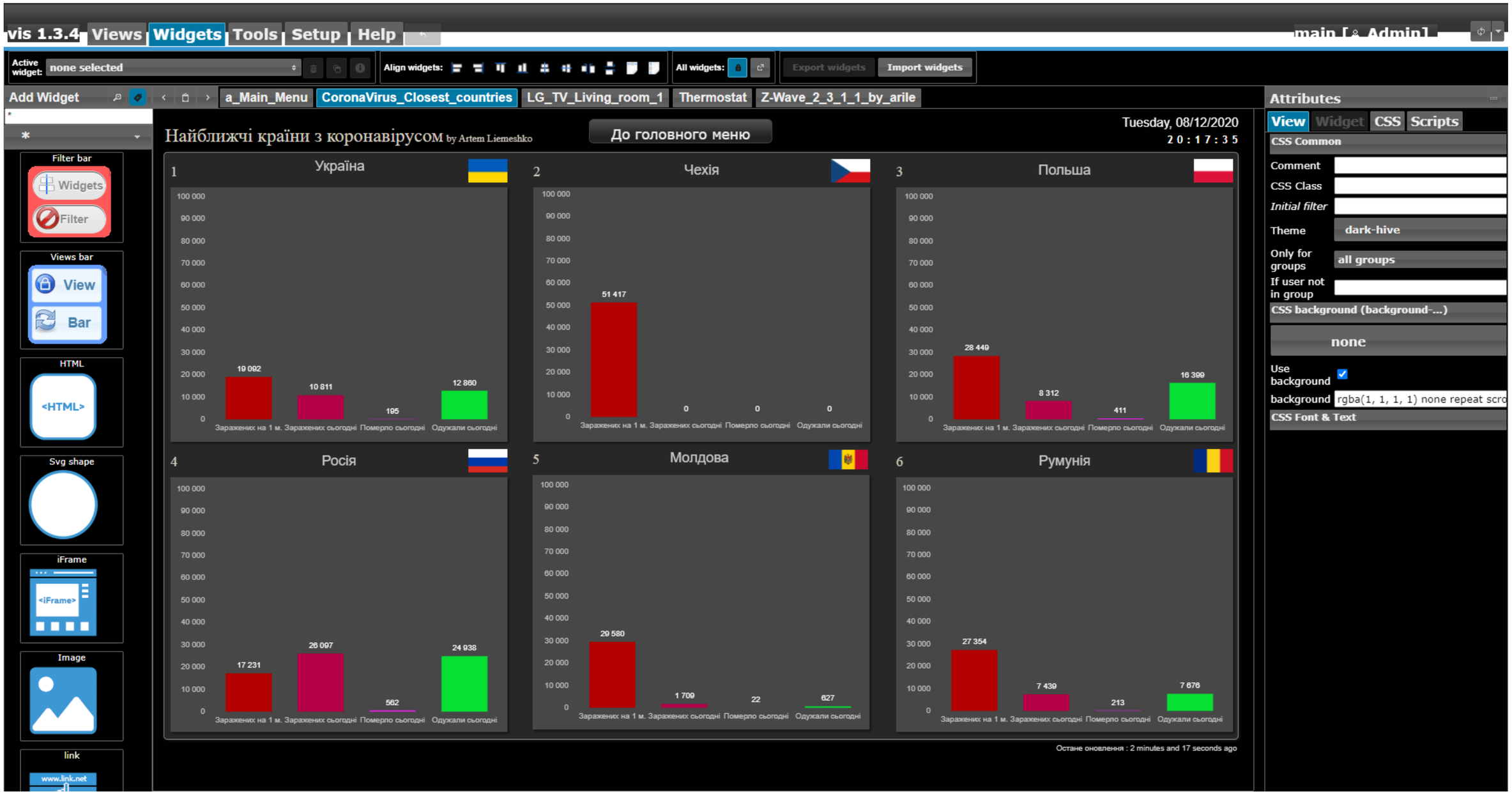
Демонстраційний плакат №2
до дипломної роботи на тему
„Система управління розумним будинком”

Розробив: Лємешко А.Д.
Прийняв: Лісовиченко О.І.

ДОДАТОК Ґ

Інтерфейс моніторингу зараження коронавірусом (coronavirus.0)

Інтерфейс моніторингу зараження коронавірусом (coronavirus.0)



Демонстраційний плакат №3
до дипломної роботи на тему
„Система управління розумним будинком”

Розробив: Лємешко А.Д.
Прийняв: Лісовиченко О.І.

ДОДАТОК Д

Результати перевірки «Unicheck»